Efficient compression of motion compensated residuals

Poh, Weekiong

Award date:
2003

Awarding institution:
University of Bath

EFFICIENT COMPRESSION OF MOTION COMPENSATED RESIDUALS

submitted by

Weekiong Poh

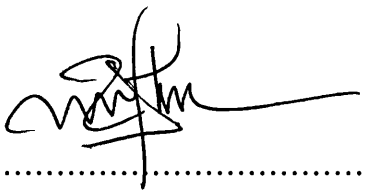
for the degree of

Doctor of Philosophy

of the

University of Bath

2003



Signed: .....

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

Prediction error images are often encountered in video processing: they include simple frame difference signals, the difference between an original signal reconstructed after down-conversion followed by up-conversion and the motion compensated prediction error in video encoding. In general these signals can be characterized as having high predominantly high frequency content and fine line structures generated by the edges of the features in the video scene. Despite the difference between these difference signals and natural images, they are often coded using a simple variant of a still image compression algorithm in many compression systems.

In this work we investigate the application of some new key technologies such as the Discrete Wavelet Transform (DWT) and Successive-Approximation Quantization to motion compensated residuals. We start by presenting an overview of the current video coding standards and implementation details of the DWT in image coding, ranging from boundary extension to fast implementations. To gain a better insight into the coding efficiency of linear transforms for residual coding, several linear transforms are evaluated from both a theoretical and a practical point of view. We then continue by exploiting some inherent characteristics of motion residuals in trying to achieve a simple and efficient motion residual coding technique. The proposed algorithm is based on bit-plane coding where its embedded

property allows precise control of coding rates since the bit-stream can be truncated at any point according to either rate or distortion constraints. The results that are reported outperform MPEG-2, H.263 and some selected wavelet-based coders.

After presenting the algorithm, we apply several techniques from our earlier works to video coding using matching pursuits. Matching pursuits is a recent advance in residual coding and produces results that are among the best in the literature. The proposed method not only improves upon the original implementation but also solves a rate/quality limiting condition known as "deadlocking".

Acknowledgements

First, I wish to express my appreciation and gratitude to my advisor, Professor D.M. Monro, for the patience and guidance he has provided me during the whole process of my graduate studies. I am also thankful to my colleagues Nick, Brian, Simon, Matt, Charith, Yanni, Neil and Sean in the Video Coding Group at the University of Bath, as well as my co-advisor Dr A.N. Evans, for their participation in numerous visual experiments. Next, I would like to thank Professor A. Mason for some intellectual discussions on residual coding and Tandberg TV for providing me with the necessary financial support over the years. Perhaps most important of all, I owe thanks to my parents for (among other things) instilling in me a desire to learn, and to my fiancée Angelyn for her unconditional love and support throughout the years.

Contents

Chapter 2: Discrete Wavelet Transform

Chapter 3: Motion-Compensated Residual Pictures

Chapter 4: Embedded DCT Coding Using Significance Block Map

Chapter 5: Improved Coding of Matching Pursuits

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The advancement of technologies in global networks and telecommunications in the last decade has opened a new era of multiple and simultaneous exchange of audio-video information. It enables users to exchange or share information using many different media: voice, audio, video, graphics, text, image, etc. Among these media, video has emerged as an information carrier that can efficiently convey a large amount of information. The cost of storage and transmitting video is however, also very high due to its inherently large requirement for data rate. The need for highly efficient compression algorithms has therefore triggered intense research both in the academic field and industry for many years.

Since the goal of video compression is to minimize communication or storage rate and hence the bandwidth or storage capacity required for a given communication system, one might suspect that the need for video compression is obviated by wideband technologies such as optical fiber communication networks, packet and digital radio, and optical storage disks, when they are increasingly employed for end-users. While it is true in some examples that available bandwidth lessens the need for complex compression algorithms, compression can still be important for a variety of reasons: More data channels can be multiplexed in wideband sys-

tems by using data compression to reduce the bandwidth required of each channel. Some data sources such as high definition television and multi-spectral data can generate data in such high volume that any transmission media, including optical fiber, can be quickly overwhelmed. Also, when a given transmission channel gradually has its spare capacity exhausted as traffic increases with time, the cost of installing an entirely new transmission facility is far greater than the cost of sophisticated terminal equipments (DTEs) which can perform efficient data compression and thereby greatly increase the limited capacity in a cost-effective manner.

To ensure interoperability between DTEs from different vendors, several international standards for audio-video compression have emerged in the last decade [44, 45, 46, 47, 49, 50, 51, 52, 53]. The organizations responsible for the development of these standards are the International Standards Organization (ISO), the International Telecommunication Union (ITU), and the International Electrotechnical Commission (IEC). Although each standard has been designed for some specific applications in mind, the encoding methods employed in these standards to exploit both the spatial and temporal redundancies within the video sequences is largely the same. Spatial redundancy is eliminated by using block-based discrete cosine transform (DCT) coding of 8×8 pixel blocks followed by scalar quantization, zig-zag scan, and variable length coding of the quantized indices. Temporal redundancy is exploited by using a block-based motion tracking technique known as motion estimation/compensation. Each frame is partitioned into equal-sized square block and each block of pixels in the current frame is matched with a similar block in the previous frame, within its immediate vicinity. The offset between the two blocks is known as a motion vector. The error between the current block and the similar block in the previous frame is then encoded using a technique similar to those used in spatial redundancy removal and transmitted along with the motion vector for the block. Using inter-frame motion compensation and block-based discrete cosine transform, current video coding techniques enable video data to be compressed by between 20 and 50 times.

While the standards have served the community well, it has been known for quite some time that motion compensated prediction error frames, also known as

displaced frame difference (DFD) or simply motion residuals, will not be optimally de-correlated by discrete cosine transform [1]. The work by Rao and coworkers has instead suggested that the discrete sine transform (DST) might be a better choice. However, as there is little or no correlation in the motion residuals (see Chapter 3) after motion compensation, the rationale behind using 2-D linear transformations, e.g. DCT, DST or DWT, for energy compaction and de-correlation seems less relevant or even counter productive.

In order to gain a better understanding of the above problem, several characteristics of the motion residuals after linear transformation will be examined in detail in this work. In particular, performance bounds, i.e. the region of achievable points in the rate distortion trade-off for certain limited statistical source, will be established for the transformed data by various linear transforms. Operational R-D curves that are obtained by using optimum bits allocation and quantizers will also be computed for each transformed data to compliment those results obtained with statistical modeling. A fast compression algorithm for motion residuals will also be investigated and its effectiveness validated in a software coding solution. In addition, we will focus on an *embedded coding technique* where the goal is to produce a so-called embedded bit-stream which has the property that the prefixes of the bit-stream yield a continuum of lower rate description of the image at the highest possible levels of quality.

1.1.1 Embedded Coding

The phrase "embedded coding" was first coined in Shapiro's landmark work [97] where the goal is to produce a so-called embedded bit-stream which has the property that the prefixes of the bit-stream yield a continuum of lower rate description of the image at the highest possible levels of quality. But the idea is not new. Similar functionality can be found in the Progressive encoding mode of JPEG [44, 120]; the first still image compression standard produced in the early 1990s by the Joint Photographic Experts Group.

Although both techniques serve the same purpose, there is one major difference that separates the two: The progressive nature of the bit-stream in [44] is created by interleaving the quantized DCT coefficients using either the spectral selection or successive approximation methods, while in [97], the embedded bit-stream is created by using a series of successively refined uniform scalar quantizers with step sizes

$$\Delta^n = 2^n, \quad n = m, m-1, \ldots, 0, \tag{1.1}$$

where m is the most significant bit-plane, in the spectral domain. The use of successively refined uniform scalar quantizers allows better rate and distortion control that is not possible by the much simpler interleaving scheme.

Another useful feature that resulted from the use of (1.1) is the implicit optimization with respect to the mean squared error (MSE). To see this point, a brief review of transform coding is needed. Figure 1.1 illustrates the structure of transform coding with scalar quantizers. Let $\mathbf{x}$ denote the sample vector $\mathbf{x} = (x_n, x_{n+1}, \ldots, x_{n+N-1})^T$ with N components. A new transformed vector $\mathbf{y}$, also with N components, is given by $\mathbf{y} = \mathbf{T}\mathbf{x}$, where $\mathbf{T}$ is a N x N invertible matrix. Each component is then quantized and the reconstructed approximation to the original vector, $\hat{\mathbf{x}}$, obtained by performing the corresponding inverse transform on the quantized transformed vector $\hat{\mathbf{y}}$. If $\mathbf{T}^T\mathbf{T} = \mathbf{T}\mathbf{T}^T = \mathbf{I}$, as is usually the case within the coding context, then MSE is equal to

$$\begin{aligned} E\left[\|\mathbf{x}-\hat{\mathbf{x}}\|^2\right] &= E\left[(\mathbf{x}-\hat{\mathbf{x}})^T(\mathbf{x}-\hat{\mathbf{x}})\right] \\ &= E\left[(\mathbf{y}-\hat{\mathbf{y}})^T\mathbf{T}\mathbf{T}^T(\mathbf{y}-\hat{\mathbf{y}})\right] \\ &= E\left[\|\mathbf{y}-\hat{\mathbf{y}}\|^2\right]. \end{aligned} \tag{1.2}$$

From (1.2) it is clear that MSE is conserved between the transform and the original domains, and if the exact value of the transform coefficient y_n is sent to the de coder, then MSE decreases by $|y_n|^2/N$. This means that to get maximum reduction in MSE, one should transmit the coefficients with the larger energy first; one way of achieving that is to use (1.1).

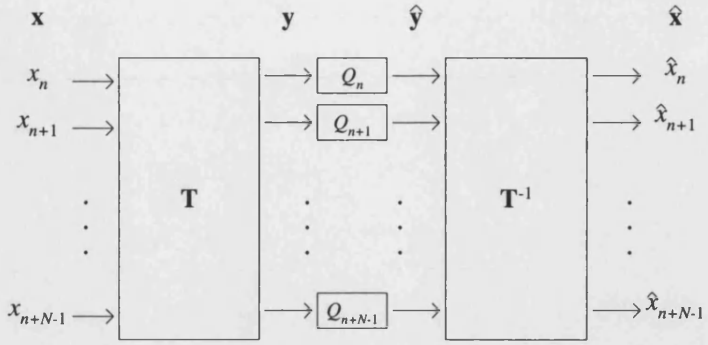


Figure 1.1 Transform coding with scalar quantization.

1.2 Current developments

In recent years, many research works have concentrated on image coding using wavelet transform. Unlike DCT, wavelet transform is a global transform where its multi-scale decomposition is well adapted at evaluating the self-similarity of a signal. More importantly, wavelets work well on a broad range of operators and lend themselves to concrete fast computational algorithms.

Of all the wavelet image compression algorithms reported in the literature, embedded solutions provide some of the most impressive compression techniques [22, 54, 61, 65, 83, 94, 97, 104, 109, 127, 130, 131] to-date. Shapiro's landmark work [97], Embedded ZeroTree Wavelet (EZW), is probably the first to utilize the multi-resolution approach to wavelets to produce a SNR scalable embedded solution. Two novel coding techniques were introduced in the author's pioneering work:

1. Multi-resolution wavelet tree structure – a method to exploit the complementary part of self-similar structures in the dyadic wavelet representation. See Figure 1.2. The basic idea is to capture cross-band correlation with the notion of coding zeros jointly.
2. Successive-approximation quantization by bit-plane encoding of the transform coefficients' magnitudes.

Both ideas have since been either used or improved by other authors with great success. Said and Pearlman [104] produced an enhanced implementation of zerotree

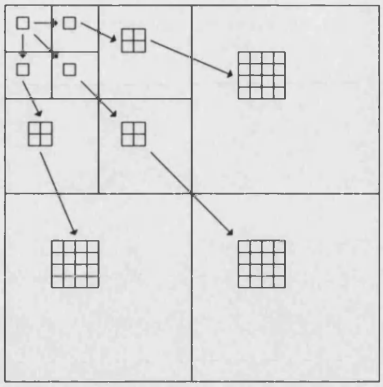


Figure 1.2 Space-frequency structure of dyadic wavelet transform.

algorithm, called Set Partitioning in Hierarchical Tree (SPIHT), that is widely considered by many as one of the best compressors in still image compression today. Xiong *et al.* [130 – 132] produced another impressive EZW variant algorithm that uses wavelet packets with explicit optimization and achieved, to the best of the author's knowledge, the best objective performance to-date. Not surprisingly, the next generation continuous-tone still image compression standard JPEG-2000 [48, 71], which is based on Taubman's work [109], also uses wavelet transform and bit-plane coding.

Despite the success in wavelet still image coding, there have been few successful attempts in wavelet video compression. Bhutani and Pearlman [13] use Shapiro's EZW algorithm to encode prediction error images obtained by recursive motion compensation and superior results to MPEG I was reported. Kim and Pearlman [56] extend SPIHT to 3-D subband video coding and superior results to MPEG-2 were also noted. Taubman and Zakhor [112] pan-shifted the video sequences before the 3-D wavelet transform using the Haar filters was applied. Sarnoff Corporation's zerotree entropy video coder (ZTE) [72], an EZW variant algorithm, has also been used successfully in the encoding of motion-compensated error frames. For low bit rates coding, works by Heising *et al.* [40], Marpe *et al.* [69, 70] and Vass *et al.* [115] have all shown encouraging results when compared to MPEG-4 and H.263 respectively.

In general, one can classify the current developments in wavelet video coding into three different groups:

1. Extensions of 2-D wavelet transform or subband coding schemes to 3-D subband coding [4, 56, 62, 112].
2. 2-D wavelet transform of the original frames followed by a multi-resolution motion-compensation in the wavelet domain [136, 137].
3. Traditional time domain motion-estimation and compensation followed by wavelet transform of motion-compensated error frames [2, 3, 6, 40, 69, 70, 72, 115].

Apart from the improved objective performance when compared against one of the current video coding standards, the subjective performance of the wavelet video coding algorithms is also usually better at low bit rates without any post-processing. Another important feature that wavelet video coding has to offer is the flexibility and ease of incorporating SNR, temporal and resolution scalability in a single bit-stream. This is a highly desirable property in some applications, e.g. video streaming, and is more readily achieved with the frameworks that exist in the first and second method, than the last one.

While wavelets have dominated much recent research works in video coding, one notable exception is the matching pursuits (MP) technique, first proposed by Mallat and Zhang [74] for signal analysis. Instead of using a 2-D linear transform, matching pursuits uses successive approximation techniques with an over-complete dictionary of prototype waveforms on the motion residuals. In fact, the traditional block-based motion-compensation technique can also be considered a particular case of matching pursuits, with a dictionary based on past frames. The basic principal of matching pursuits is very similar to that of a vector quantization scheme. Given a dictionary $D = \{\varphi_\gamma\}$ in $L^2(\mathbb{R})$, with $\|\varphi_\gamma\| = 1$ for all γ. To match an input signal $f \in L^2(\mathbb{R})$ by a linear combination of φ_γ's, start by searching for $\varphi_{\gamma 0}$ such that

$$\left|\langle \varphi_{\gamma_0}, f \rangle\right| \geq \alpha \left|\langle \varphi_n, f \rangle\right|, \quad n \neq \gamma_0, \quad 0 < \alpha \leq 1, \tag{1.3}$$

where α is usually equal to 1 and f can be expressed as its projection onto $\varphi_{\gamma 0}$ and a residual $R^1 f$,

$$f = \langle \varphi_{\gamma_0}, f \rangle \varphi_{\gamma_0} + R^1 f. \tag{1.4}$$

The algorithm is then iterated on $R^1 f$ and so on, until some convergence criterion $\|R^n f\| < \varepsilon \|f\|$, where $\varepsilon > 0$, is met. The input signal f after m decompositions can thus be written in the following compact form

$$f = \sum_{n=0}^{m-1} \langle \varphi_{\gamma n}, R^n f \rangle \varphi_{\gamma n} + R^m f. \tag{1.5}$$

By progressively isolating the structures of the signal that are coherent with respect to the chosen dictionary, an adaptive signal representation is provided in which the most significant features are extracted first. This is an extremely useful and important attribute in low bit rate coding applications.

Unfortunately, the matching pursuits algorithm involves an extremely high computational load, as well as requiring a dictionary set that is well matched to the signal characteristics. Major efforts have therefore concentrated on fast search techniques and good design of the dictionary. In an earlier work by Neff and Zakhor [78], the dictionary was designed based on the computation efficient 2-D separable Gabor functions and an ad-hoc training method was used to select the 1-D filters parameters within this restricted set. Chou *et al.* [17] optimized the dictionary for coding efficiency while Czerepinski *et al.* [16] proposed a fast implementation with a factorized dictionary that approximates a separable Gabor set. In [86], Redmill *et al.* investigated a fast two-stage filtering structure for non-separable dictionary and the same technique was later used by Neff to improve the separable Gabor dictionary [76]. The use of a non-separable dictionary is an important development, since it allows the inclusion of basis functions which provide a better match to diagonally orientated features, e.g. edges and curves, that will not be efficiently coded with a separable dictionary.

Another crucial consideration in the dictionary design is to keep the dictionary small, since a large dictionary will increase the search complexity. In a coding context however, a large dictionary allows faster convergence, i.e., fewer significant coefficients to transmit, but the downside is that a larger bit budget is needed for the definition of the waveforms that are actually used to represent the signal. To

get round this dilemma, Banham and Brailean [5] use two dictionaries: the Gabor dictionary (with different filters parameters as compared to Neff) for signals that are well localized both in space and frequency and a fast dictionary, which consists of cosine basis functions, for signals that are well modeled by a first-order stationary Markov process.

1.3 Video Compression Techniques and Standards Review

The desire of network users to have access to video of broadcast quality for their multimedia applications is tempered by the reality that internetworking bandwidth and local storage devices are of limited capacity and size. In order to store and transmit this information effectively, it has been necessary to develop techniques to reduce the average number of bits needed to represent it, that is, compress the data. Of all audio-visual compression methods that are available today, proprietary and open source alike, the core technologies employed in most systems are of the hybrid nature, the most common being the transform-prediction coding combination. In this case, the transform is used in the spatial domain, usually on a single picture, and the prediction is in the temporal domain. All these are done with one purpose in mind, to remove the substantial amount of redundant or superfluous information inherent in audio-video sources. It is only by removing the predictable or similarity component from the input signal that the data rate can be reduced.

In this section, we begin by examining some of the fundamental components that make up a compression system. We then describe several international standards for still images and video coding, and show how these compression techniques are used within the standards.

1.3.1 Basic Building Blocks

The vast majority of the designs of a lossy image compression system often follow the three components paradigm as depicted in Figure 1.3. The transformation stage mapped the input signal from one domain, usually spatial or temporal, into the fre-

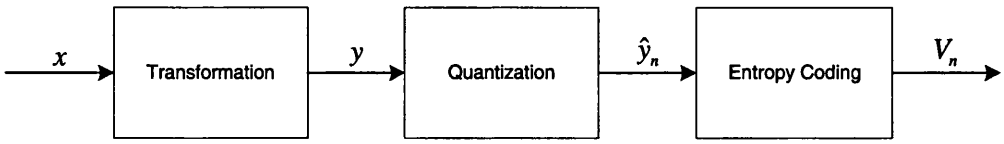


Figure 1.3 Standard model for transform coding.

quency domain. The idea of using such a transformation is that we can obtain an output vector y, often called transform coefficients or simply, coefficients, with the feature that these coefficients are much less correlated than the original samples. In addition, the information may be much more "compact" in the sense of being concentrated in only a few of the transform coefficients. Having in this sense removed redundancy, we hope as a result to be able to quantize these components more efficiently. It is important to note that there is no general "theorem" that states that uncorrelated quantities can be more efficiently quantized than can correlated variables. Another reason that supports the general approach is that models of perceptual masking of the human auditory and visual system can be incorporated into the transform domain. Roughly speaking, an adequate signal-to-noise in each frequency band is more important perceptually than an adequate overall signal-to-noise ratio where quantization noise is distributed uniformly across the frequency spectrum.

Quantization is the mapping of a large set of possible inputs R into a relatively small set of symbols $C = \{y_1,y_2,\ldots,y_N\} \subset R$. In its simplest form, a scalar quantizer (SQ) observes a single number $x \in R$ and selects the nearest approximation value from a predetermined finite set of allowed numerical values. The output values, y_i, are sometimes referred to as reconstruction thresholds or reproduction values. The most common of all SQ is the uniform quantizer, where the step size Δ, $y_i - y_{i-1} = \Delta$ for $i = 2,3,\ldots,N$, and the reconstruction thresholds for the granular cells R_i are the midpoints of the quantization interval. In the simplest application of scalar quantization to subband coefficients, one simply applies a uniform quantizer to all coefficients. The coefficients are mapped into an index labeling the interval in which it lies, and these indices are then coded using entropy code. A better alterna-

tive is to take advantage of the many zeros that appear in the quantized coefficients and of the location correspondence between those zeros.

One way to improve on simple uniform SQ is to adjust the quantizers to the different subbands. A common approach is to compute the quantization steps so as to minimize a noise power weighted by the sensitivity function of the human visual system [101, 118, 120]. This is a very important line of investigation, since perceptually based quantization will outperform quantization based on other measures by a large margin, especially at very low bit rates. Another technique of quantizer design is to rely on a simple but mathematically tractable measure, the most common of which is the mean squared error. For example, a Lloyd-Max quantizer (see Section 3.3) could be designed for each subband, based either on a training sequence of typical data or on a model of the data. Another popular design is to have a variety of SQs available for coding different coefficients within that subband. The encoder, for example, can classify the activity level of the blocks of coefficients within the band, and transmit that class information to the decoder so that it knows which SQ to use for coding those coefficients [54, 135].

The last component of a traditional coding system conserves the average bit rate by using a variable length noiseless code, also known as entropy code, which maps the quantizer output into a variable length binary index in a way that can be perfectly decoded by the receiver. The basic strategy underlying noiseless coding is to assign short code-words to a symbol or groups of symbols with high probability, and long code-words used for less probable symbols or groups of symbols, so that the average number of bits is minimized. Let $\{Y_n\}$ be a source that produces symbols from a discrete alphabet A and define the k-tuple probabilities $p(y^k) = \Pr(Y^k = y^k)$ for all $y^k = (y_0,y_1,\ldots,y_{k-1}) \in A^k$. Then, a noiseless code for Y_n is an invertible mapping of the sequence Y_n into binary symbols V_n. More specifically, noiseless codes can map individual Y_n into a varying number of bits (e.g. Huffman and Arithmetic coding) or fixed numbers of Y_n into varying number of bits (e.g. Huffman codes applied to vectors), or they can map variable numbers of Y_n into a fixed number of bits (e.g. Ziv-Lempel codes) or varying numbers of symbols into varying numbers of bits.

Shannon's theory [95] demonstrated that the smallest achievable noiseless coding bit rate for a stationary and ergodic source Y is given by the entropy rate

$$\bar{H}(Y) = \lim_{k\to\infty} H(Y^k)/k,$$
$$\text{where } H(Y^k) = -\sum_{y^k \in A^k} p(y^k)\log_2 p(y^k). \tag{1.6}$$

This optimum performance is achievable in the limit of coding arbitrarily large input vectors or permitting arbitrarily large delays in variable input length codes. If the source is a memoryless process, then $\bar{H}(Y) = H(Y_0)$, is the so-called marginal entropy. The performance of an optimum code is then bounded by $H(Y_0) \leq L < H(Y_0) + \varepsilon$, where L is the expected code length. It is important to note that simple Huffman coding on individual symbols is not guaranteed to perform close to the optimum. Only if one codes groups of increasing size can the optimum be achieved to arbitrary precision.

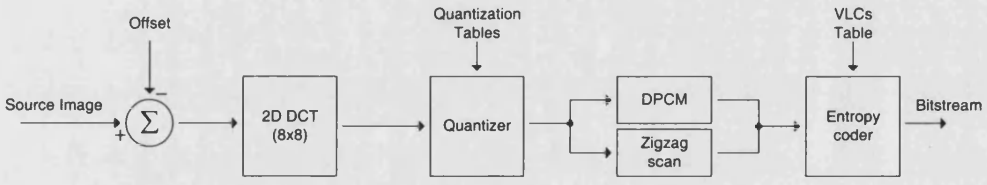
1.3.2 Still image coding: JPEG

The ISO International Standard 10918 [44] provides a standard format for compressing continuous-tone still images. This standard is commonly known by the acronym of the working group that developed the standard, the Joint Photographic Experts Group (JPEG). An overview of the standard and its scope is given in [120]. Required coding techniques for the JPEG algorithm are mainly based on 2D (8×8) DCT and either Huffman or Arithmetic coding. Four coding modes are defined in the standard:

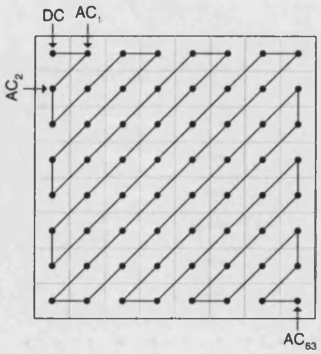
1. Sequential encoding: The image is partitioned into 8 × 8 pels block and each block is encoded in a single pass from left to right, top to bottom fashion.
2. Progressive encoding: Each 8 × 8 pels block is encoded in multiple scans, where each scan contains a partially encoded version of the image.
3. Hierarchical encoding: The image is encoded at multiple spatial resolutions so that lower resolution images can be assessed and displayed without having to decompress the full-resolution image.

4. Lossless encoding: Unlike mode 1 to 3, the lossless coding mode is based on Differential Pulse Code Modulation (DPCM) systems. This mode provides compression without any loss of quality, at the expense of considerable reduction in compression ratio.

The standard also specifies a baseline system that implements a subset of the features available in a sequential encoding mode. The block diagram depicted in Figure 1.4(a) shows the overall operation of such a system. The baseline system provides sufficient features for many general-purpose compression algorithms and has been widely adopted as the most common implementation of JPEG. The input image data are first level-shifted to a signed two's complement representation by 2^{p-1}, where p is the precision parameter of the image intensity. The level-shifted image is aggregated into blocks of 8×8 pels and the DCT is used to convert each block from the time domain into the frequency domain (of the same dimensions). This in itself does not give compression, since the information is merely represented in a different form. To achieve compression, visually tuned quantization matrices, such as those listed in Table 1.1, have been widely used to limit the accuracy



(a)



(b)

Figure 1.4 JPEG baseline encoder (a) and the zigzag scan of DCT quantized coefficients (b).

Table 1.1 JPEG quantization matrices.

Luminance

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	112	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Chrominance

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

of the transform coefficients to a level of coarseness that can be perceived by a human viewer. The quantized coefficients are then reordered in a zigzag scanning order, starting with the DC coefficient and ending with the highest frequency AC coefficient. Figure 1.4(b) illustrates the zigzag ordering of the DCT coefficients. Causal first-order prediction given by DIFF = DC_i–DC_{i-1} where DC_i and DC_{i-1} are the current and previous block DC coefficients respectively, is then used to encode the DC coefficient while all AC coefficients are converted into a set of run-length/value symbols and then encoded using Huffman variable-length codes (VLCs). The end results are a series of VLCs that describe the quantized coefficients in compress form. The whole process is then repeated for other colour components in the image.

Several extensions for the JPEG algorithm were added in 1996 to enhance its usefulness in a wide range of applications. The variable quantization extension,

for example, introduces a quantizer scale factor and provides a mean for changing the quantization matrix values at the start of any 8×8 block. This extension permits transcoding from other compression file formats such as MPEG and provides a mechanism where the masking properties of the human visual system can be exploited to improve the subjective quality. Recent standards that were born from the work of the JPEG committee include JPEG-LS [126] and JPEG-2000 [48, 71]. Both standards address a number of weaknesses and/or provide new features not available in the existing JPEG standard. The JPEG-LS standard not only provides a far superior lossless coding algorithm as compared to JPEG, but also includes a new "near-lossless" feature where each reconstructed image sample differs from the corresponding original image sample by not more than a pre-specified value.

The JPEG-2000 standard supports, among much other functionality, efficient lossy and lossless compression within a single unified coding framework. To achieve this, JPEG-2000 uses the lifting realization of a 2-channel filter bank, as depicted in Figure 1.5, which permits the implementation of both integer-to-integer and nonreversible real-to-real wavelet transforms. In the figure, $\{A_i(z)\}_{i=0}^{\lambda-1}$, $\{Q_i\}_{i=0}^{\lambda-1}$ and $\{s_i\}_{i=0}^{\lambda-1}$ denote the filter transfer functions, quantization operators and (scalar) gain, respectively. To obtain integer-to-integer mappings, the $\{Q_i\}_{i=0}^{\lambda-1}$ are selected such that they always yield integer values, and the $\{s_i\}_{i=0}^{\lambda-1}$ are chosen as integers. For real-to-real mappings, the $\{Q_i\}_{i=0}^{\lambda-1}$ are simply chosen as 1, and the $\{s_i\}_{i=0}^{\lambda-1}$ are selected from the real numbers. Since an image is a 2-D signal, the transformation stage requires the application of the filter bank in both the horizontal and vertical directions, and symmetric extension is used to facilitate filtering at the boundaries. The wavelet transform is then calculated by recursively applying the separable transform to the lowpass subband signal obtained at each level in the decomposition (see Chapter 2 for a more detailed discussion on both the wavelet transform and signal extension). For lossy compression, the standard employs a two-stage process on the wavelet coefficients for better coding efficiency. In the first stage, the wavelet coefficients are partitioned into code blocks whose dimension is a free parameter of the coding process. Each of the resulting code blocks is then independently coded using a bit-plane coder. The second stage gathers all the coding

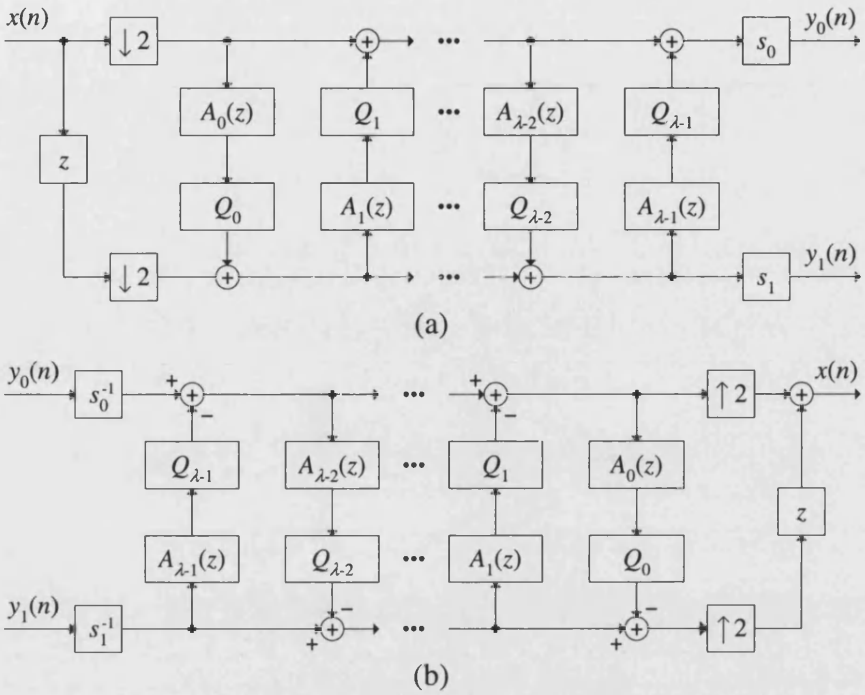


Figure 1.5 Lifting realization of 1-D 2-channel perfect reconstruction filter banks. (a) Analysis side. (b) Synthesis side.

passes, finds the optimum truncation point for each individual coding pass, and orders the truncated data according to the format (a total of 5 ordering are supported) that the user has selected. Note that if the final coded bit-stream is to have rate scalable functionality, each coding pass may have more than one truncation point. For lossless compression, the encoding procedure is identical to that of the lossy case, except that all coding passes must be coded to the end and only the integer-to-integer transform can be used.

Both standards are expected to replace the original JPEG standard, but the extent to which JPEG-2000 will enjoy success is likely to depend on its royalty-free status (both JPEG and JPEG-LS can be implemented free of royalties already).

1.3.3 Coding of Moving Pictures

Most video sequences contain high temporal redundancy; that is, successive frames within the sequence are very similar. The amount of information that needs to be

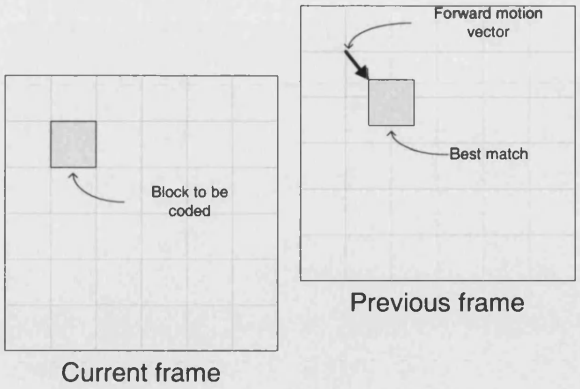


Figure 1.6 Forward motion prediction.

transmitted can therefore be reduced considerably if one encodes and transmits the difference between the current frame and previous transmitted frame, rather than by encoding the frame data itself. In practice, however, a motion estimation and compensation technique is often employed, rather than the simple DPCM scheme, because of its ability to adapt to the non-stationary nature (due to moving objects and/or camera movement) of moving sequences. To perform motion compensation, each frame is first partitioned into equal-sized square blocks and each block of pixels in the current frame is matched with a similar block in the previous frame, as illustrated in Figure 1.6. The offset between the two blocks is known as a motion vector. The error between the current block and the similar block in the previous frame is encoded and transmitted along with the motion vector for the block.

Better estimates (and hence higher compression) can be achieved by using a combination of forward motion prediction (prediction from a previous frame) and backward motion prediction (prediction from a future frame). The best match is then chosen and the appropriate motion vector encoded, or both forward and backward motion-compensated prediction may be used simultaneously to give an interpolated motion-compensated prediction. Although this technique is effective in predicting an uncovered background, the complexity of both encoder and decoder is increased: larger frame-store and frames reordering is required at both ends.

1.3.4 Videoconferencing: H.26x

Many suppliers of equipment for videoconferencing and video telephony use the compression proposed by ITU-T Recommendation H.261 [51]. The standard can be used for all classes of videoconferencing equipment with transmission in steps of 64 Kbps up to 1.92 Mbps (30 × 64 Kbps). Two image resolutions are supported: Common Intermediate Format (CIF, 352 × 288) and Quarter-CIF (QCIF, 176 × 144). Video information is represented in Y (luminance), C_r and C_b (chrominance) components. The chrominance signals are subsampled with respect to the luminance by 2:1 in both vertical and horizontal directions (also called 4:2:0 sampling).

Each frame is processed in macroblock, where a macroblock consists of four 8 × 8 pels blocks of luminance samples, one 8 × 8 blocks of C_r samples and one 8 × 8 blocks of C_b samples. Each macroblock is encoded using either an intra-coding or an inter-coding mode. The encoding process for intra-coded macroblocks is similar to the JPEG sequential coding. Each block of luminance or chrominance samples is transformed using DCT, the coefficients then quantized and variable-length encoded. Inter-coded macroblocks use the same encoding step, except that the macroblock now consists of a prediction error generated by using forward motion prediction with full pixel accuracy (no bi-directional prediction is used in H.261). In situation where motion estimation failed, that is, when the sum of the absolute difference exceeded a predefined threshold, the macroblock is intra-coded. However, if a macroblock does not change significantly with respect to the reference picture, the encoder can also choose not to encode it, and the decoder will simply repeat the macroblock located at the subject macroblock's spatial location in the reference picture.

The H.261 coder operates under a hierarchical data layer structure as shown in Figure 1.7. The four layers are Picture layer, Group of Block (GOB) layer, Macroblock (MB) layer, and the fundamental 8 × 8 pels block layer. The layers are multiplexed for transmission in series. Each layer is composed of data and the corresponding header information. One reason for the use of a hierarchical structure is

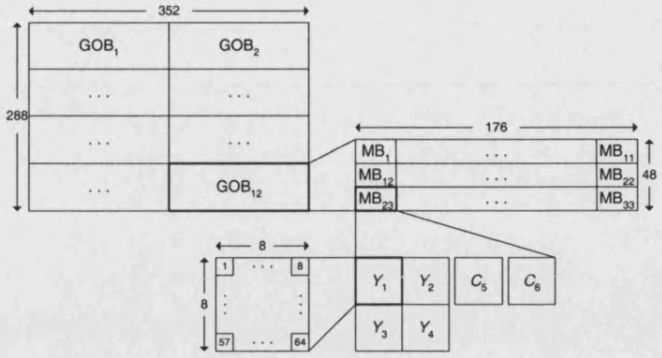


Figure 1.7 Hierarchical block structure in a CIF image.

to prevent the loss of a whole frame by using a number of hierarchical synchronizations – certain regions that have uncorrectable errors can be duplicated or interpolated.

To support low bandwidth channels such as wireless networks, public switched telephone networks (PSTN) and narrowband ISDN, ITU-T developed a new standard in 1996 known as H.263 [52] that is based on H.261. It includes several new and improved features that can provide higher quality video at low bit rate (10 – 24 kbps):

1. Supports five standardized picture formats: sub-QCIF, QCIF, CIF, 4CIF, and 16CIF.
2. Unrestricted Motion Vector Mode (Annex D): This mode allows motion vectors to point outside a picture, with edge pixels used for the prediction of nonexistent pixels.
3. Syntax-Based Arithmetic Coding Mode (Annex E): This mode allows the use of arithmetic coding instead of Huffman coding.
4. Advanced prediction mode (Annex F): This mode allows the use of overlapped block motion compensation (OBMC) with four 8×8 motion-vectors instead of 16×16 vectors per macroblock.
5. PB-Frames Mode (Annex G): In this mode, the frame structure consists of a P-picture and a B-picture that are coded together as a single PB-picture unit. The P-picture is forward predicted from the previously decoded P-picture while the B-picture, as usual, is bi-directionally predicted.

Table 1.2 12 additional new modes of H.263+.

Annex	Name	Description
I	Advanced INTRA Coding	Intra-block prediction using neighboring intra-blocks.
J	Deblocking Filter	Edge filter for 8×8 blocks.
K	Slice Structured	Grouping of MBs for better resynchronization.
L	Supplemental Enhancement info. Spec.	Format of an optional header field.
M	Improved PB-frames	Define two extra prediction modes for the B-pictures.
N	Reference Picture Selection	Allows multiple pictures for MC.
O	Temporal, SNR & Spatial scalability	Allows decoding of a sequence at more than one quality level.
P	Reference picture resampling	Supports picture warping and/or resizing.
Q	Reduced resolution update	Double the MB size so that less bits are used.
R	Independent segment decoding	Logical units are created by slices' or GOBs' boundaries at decoder.
S	Alternative INTER VLC	Allows intra VLCs to be used on inter-frames.
T	Modified Quantization	Allows non-uniform quantizers to be used.

Version 2 of H.263, also known as H.263+ in the standards community, was officially approved as a standard in January 1998 [53, 18]. It offers 12 additional new negotiable modes and features (see Table 1.2) and is backward compatible with H.263. A major improvement of H.263+ over H.263 is scalability, which can improve the delivery of video information in error-prone, packet-lossy, or heterogeneous environments by allowing multiple display rates, bit rates, and resolutions to be available at the decoder.

H.264 [49], also known as Part 10 (Advanced Video Coding) of the MPEG-4 standard, is the latest standard to emerge from ITU-T. The main objective behind the H.264 project is to develop a high-performance video coding standard by adopting a "back to basics" approach where simple and straightforward design, using well-known building blocks, is used. The new standard has a number of features that distinguish it from existing standards, while at the same time, sharing common features with other existing standards. The following are some of the key features of H.264:

1. Up to 50% in bit rate savings: Compared to H.263v2 (H.263+) or MPEG-4 Simple Profile, H.264 permits an average reduction in bit rate by up to 50% for a similar degree of encoder optimization at most bit rates.
2. High quality video: H.264 offers consistently high video quality at all bit rates, including low bit rates.
3. Adaptation to delay constraints: H.264 can operate in a low-delay mode to adapt to real-time communications applications (e.g., videoconferencing),

while allowing higher processing delay in applications with no delay constraints (e.g. video storage, sever-based video streaming applications).

4. Error resilience: H.264 provides the tools necessary to deal with packet loss in packet networks and bit errors in error-prone wireless networks.
5. Network friendliness: A new feature is the conceptual separation between a Video Coding Layer (VCL), which provides the core high-compression representation of the video picture content, and a Network Adaptation Layer (NAL), which packages that representation for delivery over a particular type of network. This facilitates easier packetization and better information priority control.

From the coding point of view, the main differences between H.264 and the other standards can be summarized as follows:

1. For the motion estimation/compensation operation, H.264 employs blocks of different sizes and shapes, higher resolution sub-pel motion estimation (e.g. ¼ and ⅛), and multiple reference frame selection.
2. H.264 uses an integer based transform that approximates the DCT transform used in previous standards, but does not have the mismatch problem in the inverse transform. The new transform is orthogonal but does not possess an orthonormal basis.
3. H.264 makes extensive use of predictive coding in areas such as intra-coding and motion vectors generation for higher coding efficiency.
4. Entropy coding can be performed using either a single Universal Variable Length Codes (UVLC) table or using Context-based Adaptive Binary Arithmetic Coding (CABAC).

Of the four differences listed, the use of integer transforms is possibly the most obvious and significant departure from previous standards. Apart from being free of mismatch problems at the inverse transform, the dynamic range of the transform operations is such that 16-bit arithmetic (for 8-bit sources) may be used throughout without any risk of overflow. The number of (integer) transforms supported by H.264 is also significantly more than the previous standards. The baseline profile of H.264 uses three transforms, depending on the type of data that is to be coded: a (hadamard) transform for the 4×4 array of luminance DC coefficients in intra macroblocks predicted in 16×16 mode, a (hadamard) transform for the 2×2 array

of chrominance DC coefficients, and a (DCT approximated) transform for all other 4 × 4 blocks. If the optional "adaptive block size transforms" mode is used, additional transforms of size 4 × 8, 8 × 4, and 8 × 8 are specified for the luminance residual (the chrominance residual decoding process remains unchanged).

It must be emphasized that although we have used the term integer transform to highlight one of the key differences between the standards, the new transform that approximates the DCT is not a "true" integer-to-integer transform. The transform is designed in such a way that only the "core" part of the transform can be carried out using integer arithmetic while the "support" part, the operation of which requires full arithmetic precision, has been "absorbed" into the quantization stage. As an example, consider the development of a 4 × 4 integer transform. The 4 × 4 DCT of an input array **X** is given by

$$\begin{aligned}\mathbf{Y} &= \mathbf{T}\mathbf{X}\mathbf{T}^T \\ &= \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \begin{bmatrix} & \mathbf{X} & \end{bmatrix} \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix},\end{aligned} \tag{1.6}$$

where

$$a = \frac{1}{2}, \quad b = \sqrt{\frac{1}{2}}\cos\frac{\pi}{8}, \quad c = \sqrt{\frac{1}{2}}\cos\frac{3\pi}{8}. \tag{1.7}$$

The matrix multiplication in (1.6) can be factorized to the following equivalent form

$$\begin{aligned}\mathbf{Y} &= \left(\mathbf{C}\mathbf{X}\mathbf{C}^T\right) \otimes \mathbf{E} \\ &= \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \begin{bmatrix} & \mathbf{X} & \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix},\end{aligned} \tag{1.8}$$

which consists of $\mathbf{C}\mathbf{X}\mathbf{C}^T$, the "core" part of the 2-D transform, and the scaling matrix (or the "support" part) **E**. The symbol ⊗ denotes scalar multiplication, the constants a and b are as before (e.g. (1.7)), and $d = c/b \cong 0.414$. To simplify the imple-

mentation of the transform, d is approximated by 0.5. The constant b, as a result, also needs to be modified so that the transform remains orthogonal:

$$b = \sqrt{2/5}. \tag{1.9}$$

The final forward transform is then given by

$$\begin{aligned}\mathbf{Y} &= \left(\mathbf{C}_f \mathbf{X} \mathbf{C}_f^T\right) \otimes \mathbf{E}_f \\ &= \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} \\ \mathbf{X} \\ \\ \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}.\end{aligned} \tag{1.10}$$

Note that the 2nd and 4th rows of matrix $\mathbf{C}_f$ are scaled by a factor of 2 (and matrix $\mathbf{E}_f$ is scaled down to compensate) so that $\mathbf{C}_f \mathbf{X} \mathbf{C}_f^T$ can be carried out using only addition, subtraction and shifts. The post-scaling operation $\otimes\mathbf{E}$ requires one multiplication for every coefficient which can be "absorbed" into the quantization process. The entire process of transform and quantization can therefore be carried out using 16-bit integer arithmetic and only a single multiply per coefficient, without any loss of accuracy. The inverse transform is given by

$$\hat{\mathbf{X}} = \mathbf{C}^T (\mathbf{Y} \otimes \mathbf{E}) \mathbf{C}. \tag{1.11}$$

1.3.5 Entertainment and Broadcast: MPEG

The first of the standards to be released by the MPEG committee was ISO 11172, commonly known as MPEG-1 [45]. The standard is optimized for coding of video and associated audio at a bit rate of about 1.5 Mbps, although data rate as high as 1.856 Mbps is also supported. The quality of the coded video is equal to that displayed on home VCR. Some examples of the use of MPEG-1 are video-on-demand applications across 1.5 Mbps links, CD-based video application, and limited resolution broadcast television.

The design of MPEG-1 is based heavily on the H.261 video coder developed for videoconferencing. The main difference between the two is the use of

half-pel accuracy and bi-directional prediction within the motion estimation and compensation process. MPEG-1 also does not support interlaced frame format typically used in broadcast television; pictures must be converted to and from the interlaced format when used in this environment. Four different types of coded picture are defined in the standard:

1. Intra-coded (I) pictures are encoded using a technique similar to the sequential mode of JPEG, without reference to any other picture. It requires the most bits to represent among the different picture types, but provides a useful access point to the sequence that is particularly important for applications that require random access.
2. Predictive (P) coded pictures are coded using motion prediction from a previous P or I picture in the sequence.
3. Bi-directionally (B) predictive coded pictures use motion prediction from a previous P or I picture or from the next P or I picture in the sequence. B pictures are not used as a reference for further predicted pictures.
4. DC (D) picture is a special case of intra in which only the DC coefficient of each 8 × 8 block is coded. D-pictures provide simple and fast forward mode but yield limited image quality.

The three main picture types, I, P and B, are grouped together in GOPs as illustrated in Figure 1.8. The structure and size of each GOP is not specified in the standard and can be chosen to suit the application. In general, a large GOP size leads to more efficient compression, since fewer I pictures are coded.

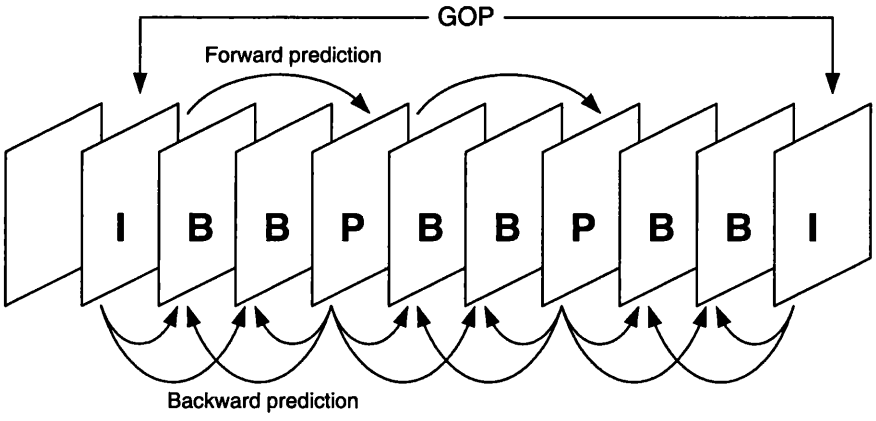


Figure 1.8 MPEG GOP structure.

Work on a new generic method for compressed representation of video sequences has been set forth with the completion of MPEG-1. The new standard is to be known as MPEG-2 [46]. It was approved as an international standard in March 1994 and is developed jointly by both ISO/IEC and ITU (Recommendation H.262). The primary application targeted during the MPEG-2 definition process was the all-digital transmission of interlaced broadcast TV quality video at coded bit rates between 4 and 9 Mbps. However, the MPEG-2 syntax has been found to be efficient for other applications such as those at higher bit rates and sample rates (e.g. HDTV). MPEG-2 Video is being defined in terms of extensible profiles, each of which will support the features needed by an important class of applications. The combination of profile and level produces an architecture that defines the ability of

Table 1.3 MPEG-2 Profiles and Levels.

Profiles

Profile	*Typical Applications*	*Features*
Simple	Broadcast	No B-pictures, No scalability, 4:2:0
Main	DSM, Broadcast	No scalability, 4:2:0
SNR scalable	ATM networks	2-layer SNR coding, 4:2:0
Spatial scalable	HDTV	2-layer SS coding, 4:2:0
High	Special applications	3-layer hybrid coding, 4:2:2

Levels

Level	*Format*	*Frame rate (Hz)*	*Compressed data rate*
Low	SIF	30	< 4 Mbits/sec
Main	CCIR-601	30	< 15 Mbits/sec
High-1440	1440x1250	60	< 60 Mbits/sec
High	1920x1250	60	< 80 Mbits/sec

Allowable combinations of profiles and levels

Level \ *Profile*	*Simple*	*Main*	*SNR scalable*	*Spatial scalable*	*High*
Low		X	X		
Main	X	X	X		X
High-1440		X		X	X
High		X			X

a decoder to handle a particular bit-stream. Table 1.3 summarized some of the most common profiles and levels, with their typical applications. Also listed in the table are the combinations (marked with an "X") that are recognized by the standard.

The MPEG-2 standard includes several enhancements and extensions to cover a wider range of applications. Note that the D-pictures that are specified in MPEG-1 for fast decoding and searching are not part of MPEG-2. The most significant enhancement over MPEG-1 is the addition of syntax for efficient coding of interlaced video (e.g. 16x8 block sizes for motion compensation, field DCT organization, Dual Prime prediction, etc). Another key features of MPEG-2 are the scalable extensions which permit the division of a continuous video signal into two or more coded bit-streams representing the video at different resolutions (spatial scalability), picture quality (SNR scalability and data partitioning), or frame rates (temporal scalability). The scalable modes of MPEG-2 offer interoperability among different services or to accommodate the varying capabilities of different receivers and network upon which a single service may operate.

MPEG-4 [50] is the next audio-visual coding standard from ISO after MPEG-1 and MPEG-2. Unlike the previous two standards, which are designed for specific applications in mind, MPEG-4 is a much broader umbrella type of standard and has a number of different technologies that are targeted at different applications. MPEG-4 was initially aimed primarily at low bit rate video communications, but its scope was later expanded for it to be much more of a multimedia coding standard. The standard provides a huge collection of new or improved technologies to fulfill the functionalities needed for clusters of applications. Eight key functionalities that are thought not to be well supported by existing or other emerging standards have been addressed. These new functionalities have been divided into three major non-orthogonal classes, based on the requirements they support.

1. Content-based interactivity: This class includes four functionalities focused on requirements for applications involving some sort of interactivity between the user and the data, namely, content-based multimedia data access tools, content based manipulation and bit-stream editing, hybrid natural and synthetic data coding, and improved temporal random access. Applications

benefiting from these functionalities include data retrieval from on-line libraries, interactive home shopping, and movie production and editing.

2. Compression: This class is composed of two functionalities: improved coding efficiency and coding of multiple concurrent data streams. They essentially aim at applications requiring an efficient storage or transmission of audio-visual information and their efficient synchronization. These functionalities will enhance some existing applications such as information browsing over Internet, and virtual reality.
3. Universal access: The remaining two functionalities are: robustness in error-prone environments and content-based scalability. These functionalities allow MPEG-4 encoded data to be accessible over a wide range of media, and with various qualities in terms of temporal and spatial resolutions for specific objects, which could be decoded by a range of decoders with different complexities. Applications benefiting from these functionalities are wireless communications, database browsing and access at different content levels, scales, resolutions, and qualities.

One key difference that differentiates the new standard from the old ones is the way scene content is constructed and transmitted. Prior to MPEG-4, video content was created from various resources such as moving video, graphics and text after they had been "composited" into a plane of pixels. These were then encoded as if they all were moving video. This is convenient but detrimental to the stated goal of having content-based interactivity. One good example that epitomizes this shortcoming is this: if one broadcaster is retransmitting another broadcaster's coverage of an event, the latter's logo cannot be removed. To get round this problem, audiovisual scenes in MPEG-4 are composed of several media objects, organized in a hierarchical fashion. At the leaves of the hierarchy, we find primitive media objects, such as

- Still images (e.g. as a fixed background);
- Video objects (e.g. a talking person - without the background);
- Audio objects (e.g. the voice associated with that person, background music);

In addition to the media objects mentioned above, MPEG-4 also defines the coded representation of objects such as text and graphics, synthetic sound, and talking synthetic heads and associated text used to synthesize the speech and animate the head. These different objects can be encoded with their own optimum coding scheme – video is coded as video, text as text, graphics as graphics – instead of treating all the pixels as moving video, and transmitted separately to the decoder in their own elementary streams. The composition only takes places after decoding instead of before encoding. See Figure 1.9. This applies for visual objects and audio alike. In order to be able to do the composition, MPEG-4 includes a special scene description language, called BIFS, for Binary Format for Scenes. The BIFS language not only describes where and when the objects appear in the scene, it can also describe behavior such as making an object spin, or two videos do a cross-fade, and conditional behavior where objects do things in response to an event such as user input.

The MPEG-4 standard consists of closely interrelated but distinct individual parts, which can be individually implemented or combined with other parts. A total of 16 parts have been added to the standard so far. The basis is formed by Systems (part 1), Visual (part 2) and Audio (part 3). We will discuss Part 2 of the standard briefly and interested readers will need to refer to the ISO documents or the MPEG website http://www.m4if.org for more details.

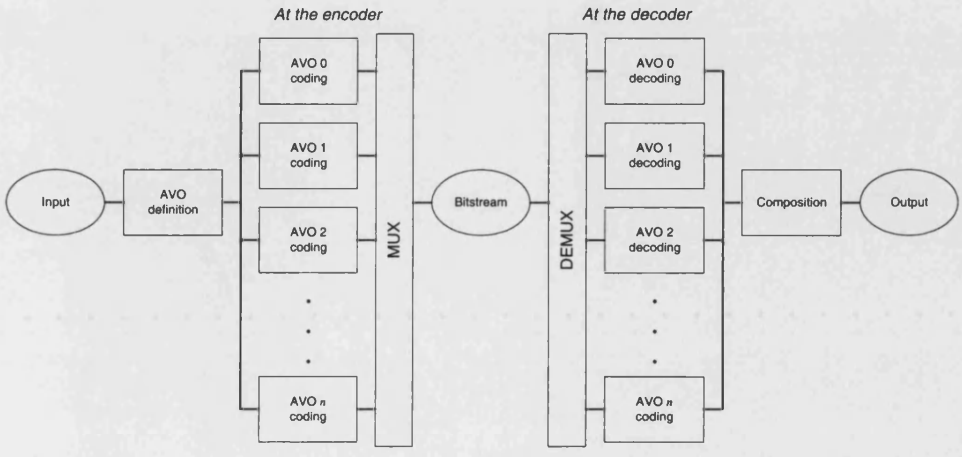


Figure 1.9 VM Encoder and Decoder Structure.

Visual (part 2 of the standard) defines an algorithm for the encoding/decoding of video objects (VOs) that can be either natural or of synthetic origin. The encoding and decoding process is carried out on the instances of VOs at a given time called Video Object Planes (VOPs). Conventional rectangular imagery (frame-based VO) is handled as a special case of such objects. The coding of conventional images and video is similar to conventional H.263 and MPEG-2 coding. It involves motion prediction/compensation followed by texture coding. For the content-based functionalities, where the image sequence input may be of arbitrary shape and location, this approach is extended by also coding shape and transparency information. Shape may be either represented by an 8-bit transparency component – which allows the description of transparency if one VO is composed with other objects – or by a binary mask.

Figure 1.10 presents a general overview of the VOP encoder structure. The same encoding scheme is applied when coding all the VOPs of a given session. The encoder consists of mainly two parts: the shape coder and the traditional motion and texture coder applied to the same VOP. The shape coder compresses the VOP's shape information, which can be either binary or of grey scale format. The methods used for binary shape information are based on block-based motion compensation and context-based arithmetic encoding. Note that no transform is used in this process. The grey scale shape information is encoded as its support function and the alpha values on the support. The support function is encoded by the binary shape coding just described and the alpha values are encoded as texture with arbitrary shape. The texture coder provides two coding modes, one for video source and the

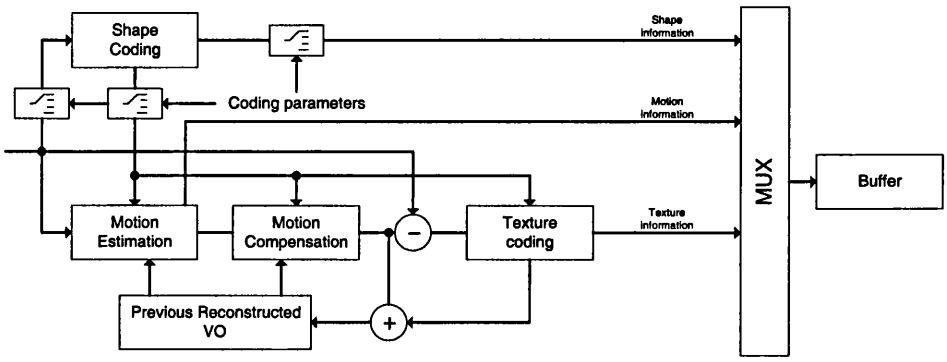


Figure 1.10 Video object encoder structure.

other for still images. Video source is encoded by the traditional DCT–Quantization–VLCs technique. The quantization process allows either H.263 or MPEG-2 quantizers to be used. A new wavelet-based zerotree variant coder that is capable of producing both SNR and spatial scalability bit-stream is used on still images. The motion coder is similar to those employed in H.263, except for the addition of some new technologies such as global motion compensation and ¼ pel accuracy.

1.4 Overview and Contributions of the thesis

Chapter 2 provides a brief description of the discrete wavelet transform (DWT) and its implementation details with finite length signal. In particular, the necessary conditions for a 2-channel perfect reconstruction FIR filter bank are studied and some of its properties discussed. The chapter is then concluded with some image coding applications using the discrete wavelet transform, as well as the wavelet packets decomposition that offers flexible time-frequency tiling and therefore better adaptivity to signal non-stationary characteristics. Chapter 3 focuses on the study of motion residuals. Several characteristics of the motion residuals are highlighted and the effect of linear transform on these images investigated. Specifically, both theoretical and operational rate-distortion performance bounds for several linear transforms are examined.

Having described the characteristics of the motion residuals, a simple and effective compression algorithm that is based on discrete cosine transform and bit-plane coding is presented in Chapter 4. The encoding of each bit-plane is driven by a significance block map that is created to take advantage of the clustering of large (or small) variance coefficients inherent in motion-compensated residuals. A simple classification scheme based on the significance block map is then applied to the identified significant coefficients to improve coding efficiency. Experimental results show that the proposed method outperforms MPEG-2 in coding of inter-frames by 0.3 – 2.5 dB. In most cases it also outperforms H.263 and SPIHT-AC and has lower complexity. Blocking artifacts are also less visible in some sequences coded at low bit rates.

Chapter 5 provides an introduction to the matching pursuits algorithm and its implementation details. A simple solution to the so-called "deadlocking" effect that is associated with using finite quantizers is described. The improved algorithm employed a precision limited quantization scheme, which was developed originally for visual consideration, and outperforms the original implementation by as much as 1.0 dB in some sequences considered.

Finally, in Chapter 6, concluding remarks for this work are made and future direction identified.

1.5 Summary

In this chapter we have discussed the motivation for this work and have provided a literature review on the works that have been done so far. Two new technologies that have dominated much research works on residual coding are wavelet transform and matching pursuits. Matching pursuits, in particular, is a recent advance which not only have been shown to produce one of the best objective performance to-date, but also produces coded video whose visual quality does not suffer from either blocking or ringing artifacts. Also described in this chapter are the main steps involved in image and video compression, concentrating in particular on the techniques adopted within the international standards for video coding. All of these techniques exploit the inherent spatial and temporal redundancy of a video sequence in order to achieve compression. The first standard to be released in this area was ISO 10918, also known as JPEG, for continuous-tone still images coding. It was later extended by both ISO/IEC and ITU-T to form the next generation video coding standards, e.g. H.261, H.263, H.264, MPEG-1, MPEG-2 and MPEG-4, which are in use today. H.261 and H.263 were designed primarily for low bit rate applications while MPEG-1 and MPEG-2 have been developed for mid-to-high bit rate applications such as home entertainment and broadcast applications. Both H.264 and MPEG-4, on the other hand, are designed as a generic coding standard to support a wide range of applications, including those that are not well supported by the previous standards.

Chapter 2

Discrete Wavelet Transform

2.1 Introduction

In recent years, wavelets and wavelet techniques have generated much interest, both in applied areas as well as in more theoretical ones. Wavelet theory involved representing general functions in terms of simpler, fixed building blocks at different scales and position, much as the Fourier transform represents signals in terms of elementary periodic waves. The terms in a wavelet expansion are built out of dilates and shifts of a single "mother wavelet" so the expansion provides better time-frequency selectivity as compared to others. This makes wavelet transform better suited at coding and analysis of nonstationary or aperiodic signals. In signal processing, fast wavelet transform is known as subband coding; with structure proposed by Mallat [66] and with appropriately designed digital filters, it can be related to continuous wavelet series expansion.

In this chapter, we first provide a simple mathematical analysis of filter banks and show how it is related to a discrete-time wavelet transform (DWT). We then review some properties of DWT and theirs role from a coding point of view. An elegant generalization of wavelet transform is then discussed, and finally, implementation issues with finite-length signals are addressed.

2.2 Filter Banks

Filter banks are efficient convolution structures that have been extensively used in subband coders for speech, and in transmultiplexers (devices to convert time-division multiplexed data to frequency-division multiplexed data and vice versa). In a filter bank, a data sequences $x(n)$ is decomposed into M components, each one carrying a single frequency subband of the original signal. It is desirable to design the filter bank such that these subbands can be recombined to recover the original signal (perfect reconstruction). Figure 2.1 shows the classical structure of a one-dimensional filter bank. The first process is called analysis, while the second is called synthesis. The output of the analysis is referred to as the subband signal, with as many subbands as there are filters in the filter bank. In multi-rate digital signal processing, the down-sampler and up-sampler operators play a fundamental role, reducing and increasing, respectively, the sampling frequency by an integer factor. If the filters are assumed to be ideal and the bandpass signal in a subband has a bandwidth of π/M, it can be down-sampled by a factor $M : 1$ without loss of information, making this the principle of critically (or maximally) decimated filter banks. On the synthesis side, the subband signal is up-sampled by a factor of $1 : M$, filtered to cancel imaging, and the subbands contributions are, thus, summed to recover the original signal.

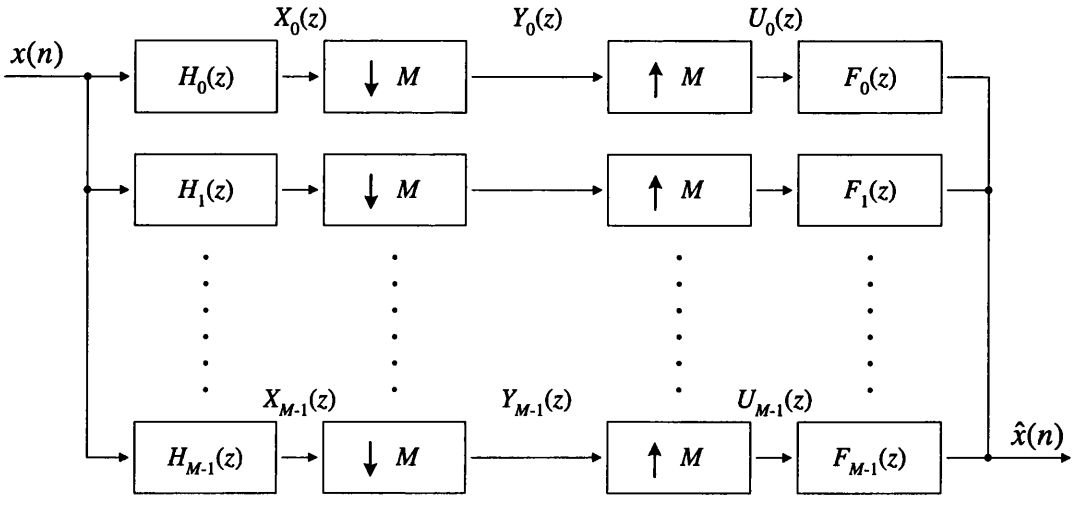


Figure 2.1 An M-channel critically decimated filter bank.

From Figure 2.1, the input sequence $x(n)$ is convolved with the analysis filters to give

$$X_k(z) = H_k(z)X(z), \quad 0 \le k \le M-1, \tag{2.1}$$

which is then down-sampled by a factor of M to produce $y_k(n)$. Thus,

$$Y_k(z) = \frac{1}{M}\sum_{l=0}^{M-1} H_k(z^{1/M}W^l)X(z^{1/M}W^l), \tag{2.2}$$

where

$$W = e^{-j2\pi/M}. \tag{2.3}$$

Note that although filtering operation on $x(n)$ is linear and time-invariant, the down-sampling operation in combination with the filtering results in a time-variant system. To understand the construction of (2.2), it is convenient to define a sequence $\tilde{x}_k(n)$ as

$$\tilde{x}_k(n) = \begin{cases} x_k(n), & n = 0, \pm M, \pm 2M \\ 0, & otherwise. \end{cases} \tag{2.4}$$

Clearly, $\tilde{x}_k(n)$ can be viewed as a sequence obtained by multiplying $x_k(n)$ with a periodic train of impulses $p(n)$, with period M, where its discrete Fourier series representation is given by

$$p(n) = \frac{1}{M}\sum_{l=0}^{M-1} e^{j2\pi nl/M}. \tag{2.5}$$

Therefore,

$$y_k(n) = \tilde{x}_k(nM) = x_k(nM)p(nM) = x_k(nM). \tag{2.6}$$

By making use of the relations in (2.5) and (2.6) and taking the z-transform of the output sequence $y_k(n)$, we obtain

$$\begin{aligned} Y_k(z) &= \sum_{n=-\infty}^{\infty} \tilde{x}_k(nM)z^{-n} \\ &= \sum_{n=-\infty}^{\infty} \tilde{x}_k(n)z^{-n/M} \\ &= \sum_{n=-\infty}^{\infty} x_k(n)\left[\frac{1}{M}\sum_{l=0}^{M-1} e^{j2\pi nl/M}\right]z^{-n/M} \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{M}\sum_{l=0}^{M-1}\sum_{n=-\infty}^{\infty} x_k(n)(e^{-j2\pi l/M} z^{1/M})^{-n} \\
&= \frac{1}{M}\sum_{l=0}^{M-1} X_k(e^{-j2\pi l/M} z^{1/M}) \\
&= \frac{1}{M}\sum_{l=0}^{M-1} H_k(e^{-j2\pi l/M} z^{1/M}) X(e^{-j2\pi l/M} z^{1/M}),
\end{aligned}
\tag{2.7}
$$

where the last step makes use of (2.1). Now the subband output sequence is up-sampled by a factor of M to produce $u_k(n)$ where its z-transform is given by

$$
\begin{aligned}
U_k(z) &= Y_k(z^M) \\
&= \frac{1}{M}\sum_{l=0}^{M-1} H_k(zW^l) X(zW^l).
\end{aligned}
\tag{2.8}
$$

$U_k(z)$ is then convolved with the synthesis filters and sum to produce the reconstructed signal

$$
\begin{aligned}
\hat{X}(z) &= \sum_{k=0}^{M-1} F_k(z) U_k(z) \\
&= \frac{1}{M}\sum_{l=0}^{M-1} X(zW^l)\sum_{k=0}^{M-1} H_k(zW^l) F_k(z) \\
&= \sum_{l=0}^{M-1} A_l(z) X(zW^l),
\end{aligned}
\tag{2.9}
$$

where

$$
A_l(z) = \frac{1}{M}\sum_{k=0}^{M-1} H_k(zW^l) F_k(z), \quad 0 \le l \le M-1. \tag{2.10}
$$

It is obvious from the above discussion that both decimation and interpolation operations introduced aliasing ($X(zW^l)$, $l > 0$) in the reconstructed signal. To eliminate aliasing, one requires

$$
A_l(z) = 0, \quad 1 \le l \le M-1. \tag{2.11}
$$

For the filter banks to achieve perfect reconstruction, i.e. the reconstructed signal to be free from amplitude and phase distortions, both analysis and synthesis filters must satisfy

$$
T(z) = A_0(z) = \frac{1}{M}\sum_{k=0}^{M-1} H_k(z) F_k(z) = cz^{-n_0}, \quad c \neq 0. \tag{2.12}
$$

For $M = 2$, (2.9) can be written as

$$\hat{X}(z) = T(z)X(z) + A(z)X(-z), \tag{2.13}$$

where

$$\begin{aligned} T(z) &= \frac{1}{2}\{H_0(z)F_0(z) + H_1(z)F_1(z)\} \\ A(z) &= \frac{1}{2}\{H_0(-z)F_0(z) + H_1(-z)F_1(z)\} \end{aligned} \tag{2.14}$$

must satisfy (2.11) and (2.12) for perfect reconstruction. Hence,

$$H_0(-z)F_0(z) + H_1(-z)F_1(z) = 0 \tag{2.15}$$

$$H_0(z)F_0(z) + H_1(z)F_1(z) = 2z^{-n_0}. \tag{2.16}$$

By choosing $F_0(z) = H_1(-z)$ and $F_1(z) = -H_0(-z)$, condition (2.15) is satisfies automatically. We are then left with the design of H_0 and H_1. Two possibilities that dominated much earlier works [23, 103, 113, 117] are

$$H_1(z) = H_0(-z) \tag{2.17}$$

and

$$H_1(z) = -z^{-N+1}H_0(-z^{-1}), \tag{2.18}$$

where N is the filter length. Using (2.14) in (2.17), the polyphase representation of the analysis filters and the transfer matrix are given by

$$\begin{aligned} H_0(z) &= E_0(z^2) + z^{-1}E_1(z^2) \\ H_1(z) &= E_0(z^2) - z^{-1}E_1(z^2) \\ T(z) &= 2z^{-1}E_0(z^2)E_1(z^2). \end{aligned} \tag{2.19}$$

If $H_0(z)$ is FIR, so is $E_0(z)$, $E_1(z)$ and $T(z)$. Since PR requires $T(z) = cz^{-n_0}$, $E_0(z)$ and $E_1(z)$ must also be a delay, that is, $E_0(z) = c_0 z^{-n_0}$ and $E_1(z) = c_1 z^{-n_1}$. As a result, the analysis filters are restricted to two coefficients and therefore cannot have sharp cutoff and good stopband attenuation. Note that we could have chosen $E_1(z) = 1/E_0$ but the filter becomes IIR. In conclusion, using (2.17) in filter banks design eliminates aliasing and phase distortion but not amplitude distortion, unless the filters have the simple form just discussed. Therefore, one can only minimize the distortion such that

$$\left|H_0(e^{j\omega})\right|^2 + \left|H_1(e^{j\omega})\right|^2 \cong 1. \tag{2.20}$$

Filter banks that use (2.17) are also commonly known as QMF (Quadrature Mirror Filter). Now using (2.18) in (2.16), the perfect reconstruction condition simplifies to

$$P(z)+P(-z)=2, \tag{2.21}$$

where $P(z) = H_0(z^{-1})H_0(z)$ is a zero-phase half band FIR filter whose impulse response satisfies the condition

$$h(2n)=\begin{cases}1, & n=0\\ 0, & otherwise.\end{cases} \tag{2.22}$$

Note that the zero-phase requirement implies $h(n) = h(-n)$. On the unit circle, the filter satisfies the power complementary condition

$$\left|H_0(e^{j\omega})\right|^2+\left|H_0(e^{j(\omega+\pi)})\right|^2=2, \tag{2.23}$$

that is, its magnitude squared summed up to a constant. Filters satisfying (2.21) yield orthogonal filter banks with perfect reconstruction. The flattest $P(z)$ will lead us to the famous Daubechies wavelet [23]. As an example, we will construct Daubechies' D_6 (of length 6) filter. Assume the minimum-degree solution. Then $P(z)$ has powers of z going from $(-2k + 1)$ to $(2k - 1)$ and has the form [117]

$$P(z)=(1+z^{-1})^k(1+z)^k R(z), \tag{2.24}$$

where $R(z)$ is symmetric $(R(z^{-1}) = R(z))$ and has powers of z going from $(-k + 1)$ to $(k - 1)$. For D_6 filter, $k = 3$ and

$$R(z)=az^2+bz+c+bz^{-1}+az^{-2}. \tag{2.25}$$

Substituting (2.25) into (2.24) we obtain

$$\begin{aligned}P(z)=&\,az^5+(b+6a)z^4+(c+6b+15a)z^3+(16b+6c+20a)z^2+\\&(16a+26b+15c)z+20c+30b+12a+(16a+26b+15c)z+\\&(16b+6c+20a)z^{-2}+(c+6b+15a)z^{-3}+(b+6a)z^{-4}+az^{-5}.\end{aligned}$$

Equating even powers of z with 0 and z^0 with 1 (see (2.22)) yield a system of linear equations

$$\begin{aligned}b+6a&=0\\16b+6c+20a&=0\\20c+30b+12a&=1,\end{aligned}$$

Table 2.1 Matlab codes for finding Daubechies filters using ceptral analysis.

```
Function H = dauf(L)

% L == filter length.

HL = L/2;
N  = 8*L;

z = exp(j*2*pi*[0:N-1]/N);

P = zeros(1,N);
for l=0:HL-1, P = P + nchoosek(HL+l-1,l) * (-z).^l .* ((1 - z.^(-1))/2).^(2*l); end
P = 4*P;

iLP = ifft(log(P));
iLP(N/2+1:N) = zeros(1,N/2);    % Retain just the causal part.
iLP(1) = iLP(1)/2;              % Value at zero is shared between
                                % the causal and anticausal part.
MPS = exp(fft(iLP,N));          % Min phase spectral factor of P.

H = real(ifft(MPS .* ((1 + z.^(-1))/2).^HL));
H = H(1:L)/sqrt(2);
```

which solution is given by

$$a = \frac{3}{256}, \quad b = -\frac{9}{128}, \quad c = \frac{19}{128}.$$

Solving (2.25), we find 4 zeros at $\{2.7127 \pm j1.4439, 0.2873 \pm j0.1529\}$. Now construct $H_0(z)$ (with appropriate scaling) by taking all roots inside the unit circle, e.g. $0.2873 \pm j0.1529$, and also take one out of every double root on the unit circle (from (2.24)):

$$H_0(z) = 0.3327 + 0.8070z^{-1} + 0.4598z^{-2} - 0.1351z^{-3} - 0.0855z^{-4} + 0.0352z^{-5}.$$

Note that the resulting filters produced by the above roots finding technique are usually only approximately orthogonal as the filter size grows. For very long filters, the Ceptral method (see Table 2.1 for the Matlab codes) is often used because it goes directly to the spectral factor $H_0(z)$, without the zeros. It is based on splitting $\log P(z)$ into $\log H_0(z) + \log H_0(z^{-1})$.

2.3 Wavelet Theory and its Relation to Filter Banks

A generic method of signal analysis is to represent the signal by a set of basis functions whose properties are well understood. For stationary signal analysis, the best-known transform is the Fourier transform pair which is given by

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \tag{2.26}$$

$$f(t) = \int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega. \tag{2.27}$$

The information provided by the integral corresponds to all time instances, since the integration is from minus infinity to plus infinity over time. Any abrupt change in time is reflected in the whole spectrum, and therefore, the Fourier transform is not suitable if the signal has time-varying frequency. To overcome this drawback, one can apply a time window on the data and then taking the Fourier transform. This is known as the Short-Time-Fourier-Transform (STFT) and can be calculated as

$$STFT(\omega,l) = \int_{-\infty}^{\infty} f(t)w^*(t-l)e^{-j\omega t}dt. \tag{2.28}$$

The disadvantage with STFT is the fact that its roots go back to what is known as the Heisenberg's Uncertainty Principle. This principle, which originally applied to the momentum and location of moving particles, can be applied to the time-frequency information of a signal. Simply, this principle states that one cannot know the exact time-frequency representation of a signal, i.e., one cannot know what spectral components exist at what instances of times. What one can know are the time intervals in which certain bands of frequencies exist, which is a resolution problem. If the frequency components are well separated from each other in the original signal, then we may sacrifice some frequency resolution and go for good time resolution i.e. a short time window, since the spectral components are already well separated from each other. However, if this is not the case, then it is very difficult to find a good window function.

Wavelet transform (WT) provides a compromise on the time-frequency problem. WT analyses the signal at different frequencies with different resolutions

– good time resolution and poor frequency resolution at high frequencies and good frequency resolution and poor time resolution at low frequencies. Every spectral component is not resolved equally as was the case in the STFT. The continuous wavelet transform (CWT) is defined as [23]

$$C(a,b) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}(t)dt, \tag{2.29}$$

where

$$\psi_{a,b}(t) = a^{-\frac{1}{2}}\psi\left(\frac{t-b}{a}\right) \tag{2.30}$$

is a basis function called *wavelet* (e.g. small wave) and can be viewed as a dilated and translated version of the *mother wavelet* $\psi(t)$. The arguments a and b denote the *scale* and *translation* respectively. The term translation is used in the same sense as it was used in the STFT. It is related to the location of the window, as the window is shifted through the signal. This term obviously corresponds to time information in the transform domain. However, we do not have a frequency parameter, as we had in the STFT. Instead we have a scale. Scaling, as a mathematical operation, either dilates or compresses a signal. Smaller scales correspond to dilated (or stretched out) signals and large scales correspond to compressed signals. However, in the definition of the wavelet transform, the scaling term is used in the denominator, and therefore, the opposite of the above statements holds. The relation between scale and frequency is that low scales correspond to high frequencies and high scales to low frequencies. The inverse transform is given by the relationship

$$f(t) = \frac{1}{K_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} C(a,b)\psi_{a,b}(t)\frac{dadb}{a^2}. \tag{2.31}$$

The constant K_ψ depends only on $\psi(t)$ and is given by

$$K_\psi = \int_{-\infty}^{\infty}\frac{|\hat{\psi}(\omega)|^2}{|\omega|} < \infty, \tag{2.32}$$

where $\hat{\psi}(\omega)$ is the Fourier transform of $\psi(t)$.

For wavelet theory to be useful, it must come with fast algorithms for machine computation, that is, a method like FFT both for finding the wavelet coefficients $C(a,b)$ and for reconstructing the function they represent. The multiresolution

approach, pioneered by Mallet [66], offers this possibility, by linking wavelet to the subband decompositions of signals. A multiresolution analysis consists of a sequence of embedded closed subspaces [23, 66]

$$\ldots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \ldots \tag{2.33}$$

such that

- *Completeness*:

$$\overline{\bigcup_{m\in\mathbb{Z}} V_m} = L^2(\mathbb{R}) \text{ and } \bigcap_{m\in\mathbb{Z}} V_m = \{0\}. \tag{2.34}$$

- *Scale Invariance*:

$$f(t)\in V_m \Leftrightarrow f(2^m t)\in V_0. \tag{2.35}$$

- *Shift Invariance*:

$$f(t)\in V_0 \Rightarrow f(t-n)\in V_0 \text{ for all } n\in\mathbb{Z}. \tag{2.36}$$

- *Existence of a Basis*: There exist $\varphi \in V_0$, such that

$$\{\varphi(t-n)\,|\,n\in\mathbb{Z}\} \tag{2.37}$$

is an orthonormal basis for V_0, whose orthonormality implies

$$\delta(n) = \langle\varphi(t),\varphi(t-n)\rangle \Leftrightarrow \sum_k \left|\hat{\varphi}(\omega+2\pi k)\right|^2 = 1, \tag{2.38}$$

where $\hat{\varphi}(\omega)$ is the Fourier transform of $\varphi(t)$. Using (2.35 – 2.37), one obtains that $\{2^{-m/2}\varphi(2^{-m}t-n)\,|\,n\in\mathbb{Z}\}$ is a basis for V_m. The functions $\{\varphi_{m,n}\}$ are known as scaling functions since they build up scaled versions of functions in $L^2(\mathbb{Z})$. Any function $f(t)$ can be expressed as the limit of the approximation $f^m(t) \in V_m$ for m tends to $-\infty$, i.e.

$$f(t) = \lim_{m\to-\infty} f^m(t). \tag{2.39}$$

Since $\varphi \in V_0 \subset V_{-1}$, and $\varphi_{-1,n}$ are an orthonormal basis in V_{-1}, we have

$$\varphi(t) = \sqrt{2}\sum_n h_0(n)\varphi(2t-n), \tag{2.40}$$

where $h_0(n) = \langle\varphi,\varphi(2t-n)\rangle$. Taking the Fourier transform of both sides, we obtain

$$\begin{aligned}\hat{\varphi}(\omega) &= \sqrt{2}\int\sum_n h_0(n)\varphi(2t-n)e^{-j\omega t}dt \\ &= \frac{\sqrt{2}}{2}\sum_n h_0(n)\int\varphi(t)e^{-j\omega t/2}e^{-j\omega n/2}dt \\ &= \frac{1}{\sqrt{2}}\sum_n h_0(n)e^{-j(\omega/2)n}\int\varphi(t)e^{-j(\omega/2)t}dt \\ &= \frac{1}{\sqrt{2}}H_0(\omega/2)\hat{\varphi}(\omega/2),\end{aligned} \tag{2.41}$$

where

$$H_0(\omega/2) = \sum_n h_0(n)e^{-j(\omega/2)n}.$$

Substituting (2.41) into (2.38) for 2ω gives

$$\left|H_0(\omega)\right|^2 + \left|H_0(\omega+\pi)\right|^2 = 2, \tag{2.42}$$

the condition on $\{h_0(n)\}$ for $\{\varphi(t\text{-}n)\}$ to be orthogonal. Note that (2.42) was already given in (2.23) (a hint that there is a strong connection between the two).

The basic tenet of multiresolution analysis is that whenever a collection of closed subspaces satisfies (2.33 – 2.38), then there exists an orthonormal wavelet basis for $L^2(\mathbb{R})$:

$$\upsilon_{m,n}(t) = 2^{-m/2}\upsilon(2^{-m}t-n), \quad m,n\in\mathbb{Z}, \tag{2.43}$$

such that $\{\upsilon_{m,n}\}$, $n\in\mathbb{Z}$ is an orthonormal basis for W_m, where W_m is the orthogonal complement of V_m in $V_{m\text{-}1}$, i.e. $V_{m-1} = V_m \otimes W_m$. For proof, see [23, 117]. Any function $f(t)$ in $L^2(\mathbb{R})$ can now be represented as

$$P_{m-1}f(t) = P_m f(t) + \sum_{n\in\mathbb{Z}}\langle f(t),\upsilon_{m,n}\rangle\upsilon_{m,n}, \tag{2.44}$$

where P_m is the orthogonal projection onto V_m. The aim is to find $\upsilon(t) \in W_0$, such that the $\upsilon(t-n)$, $n\in\mathbb{Z}$, constitute an orthonormal basis for W_0. Since $\upsilon\in W_0 \subset V_{-1}$,

$$\upsilon(t) = \sqrt{2}\sum_n h_1[n]\varphi(2t-n), \tag{2.45}$$

where $h_1(n) = \langle\upsilon,\varphi(2t-n)\rangle$. Taking the Fourier transform one obtains

$$\hat{\upsilon}(\omega) = \frac{1}{\sqrt{2}} H_1(\omega/2)\hat{\varphi}(\omega/2), \tag{2.46}$$

where $H_1(\omega)$ is a 2π-periodic function from $L^2([0, 2\pi])$. The fact that $\upsilon(t)$ belongs to W_0, which is orthogonal to V_0, implies that

$$\langle \varphi(t-k), \upsilon(t) \rangle = 0, \text{ for all } k.$$

This can be expressed as (in the Fourier domain)

$$\int \hat{\upsilon}(\omega)\hat{\varphi}^*(\omega)e^{j\omega k} d\omega = 0,$$

or equivalently,

$$\int_0^{2\pi} e^{j\omega k} d\omega \sum_l \hat{\upsilon}(\omega + 2\pi l)\hat{\phi}^*(\omega + 2\pi l) = 0;$$

hence

$$\sum_l \hat{\upsilon}(\omega + 2\pi l)\hat{\phi}^*(\omega + 2\pi l) = 0. \tag{2.47}$$

Substituting (2.41) and (2.46) into (2.47), regrouping the sums for odd and even l, and using (2.38) gives ($\Omega = \omega / 2$)

$$H_1(\Omega)H_0^*(\Omega) + H_1(\Omega + \pi)H_0^*(\Omega + \pi) = 0. \tag{2.48}$$

Since $H_0^*(\omega)$ and $H_0^*(\omega + \pi)$ cannot go to zero at the same time (see (2.42)), this implies the existence of a 2π-periodic function $\lambda(\omega)$ so that

$$H_1(\omega) = \lambda(\omega)H_0^*(\omega + \pi)$$

and

$$\lambda(\omega) + \lambda(\omega + \pi) = 0.$$

Choosing $\lambda(\omega) = -e^{-j\omega}$ gives

$$H_1(\omega) = -e^{-j\omega}H_0^*(\omega + \pi), \tag{2.49}$$

or, in time-domain

$$h_1(n) = (-1)^n h_0(-n + 1),$$

which is the same as (2.18). Finally, the wavelet is obtained as

$$\hat{\upsilon}(\omega) = -\frac{1}{\sqrt{2}} e^{-j\omega/2} H_0^*(\omega/2 + \pi)\hat{\varphi}(\omega/2), \tag{2.50}$$

$$\upsilon(t) = \sqrt{2}\sum_n (-1)^n h_0(-n+1)\varphi(2t-n).$$

To evaluate (2.44), first let $g(n) = (-1)^n h_0(-n+1)$ and rewrite (2.43) as

$$\begin{aligned}\upsilon_{m,k}(t) &= 2^{-m/2}\upsilon(2^{-m}t-k)\\ &= 2^{-m/2}\sum_n 2^{1/2} g(n)\varphi(2^{-m+1}t-2k-n)\\ &= \sum_n g(n)\varphi_{m-1,2k+n}(t)\\ &= \sum_n g(n-2k)\varphi_{m-1,n}(t).\end{aligned} \tag{2.51}$$

It follows that

$$\langle f(t), \upsilon_{1,k}\rangle = \sum_n g^*(n-2k)\langle f(t), \varphi_{0,n}\rangle,$$

i.e., the $\langle f(t), \upsilon_{1,k}\rangle$ are obtained by convolving the sequence $\left(\langle f(t), \varphi_{0,n}\rangle\right)_{n\in\mathbb{Z}}$ with $\left(g^*(-n)\right)_{n\in\mathbb{Z}}$ and then retaining only the even samples. Similarly, we have

$$\langle f(t), \upsilon_{m,k}\rangle = \sum_n g^*(n-2k)\langle f(t), \varphi_{m-1,n}\rangle \tag{2.52}$$

and from (2.40)

$$\begin{aligned}\varphi(t) &= 2^{-m/2}\varphi(2^{-m}t-n)\\ &= \sum_n h_0^*(n-2k)\varphi_{m-1,n}(t),\end{aligned} \tag{2.53}$$

hence

$$\langle f(t), \varphi_{m,k}\rangle = \sum_n h_0^*(n-2k)\langle f(t), \varphi_{m-1,n}\rangle. \tag{2.54}$$

The whole process can be viewed as the computation of successively coarser approximations of $f(t)$, together with the difference in "information" between every two successive levels. In this view we start out with a fine-scale approximate to $f(t)$, $f^0(t) = P_0 f(t)$, and we decompose $f^0(t) \in V_0 = V_1 \otimes W_1$ into $f^0(t) = f^1(t) + \delta^1(t)$, where $f^1(t) = P_1 f^0(t) = P_1 f(t)$ is the next coarser approximation of $f(t)$, and $\delta^1(t) = f^0(t) - f^1(t) = Q_1 f^0(t) = Q_1 f(t)$ where Q_m is the orthogonal projection onto W_m. Stated explicitly,

$$\begin{aligned}f^{m-1}(t) &= f^m(t) + \delta^m(t)\\ &= \sum_k c_k^m \varphi_{m,k} + \sum_k d_k^m \upsilon_{m,k},\end{aligned} \tag{2.55}$$

where

$$\begin{aligned} c_k^{m-1} &= \langle f^{m-1}(t), \varphi_{m-1,n} \rangle \\ &= \sum_k c_k^m \langle \varphi_{m,k}, \varphi_{m-1,n} \rangle + \sum_k d_k^m \langle \upsilon_{m,k}, \upsilon_{m-1,n} \rangle \\ &= \sum_k \left[h_0(n-2k) c_k^m + g(n-2k) d_k^m \right] \end{aligned} \tag{2.56}$$

is equivalent to the synthesis step of a filter bank.

2.4 Regularity and Vanishing Moments

Traditional subband coding attempted to improve the coding performance by optimizing various filter properties such as stopband attenuation, phase linearity, coding gain, etc. Wavelet theory requires another constraint – the basis function should be *regular*. In simple terms, regularity is a smoothness requirement on the scaling function, and can be mathematically defined as continuity of this function and its derivatives. The scaling function (2.41) after N iterations becomes

$$\hat{\varphi}(\omega) = M_0(\omega/2) M_0(\omega/4) \ldots M_0(\omega/2^N) \hat{\varphi}(\omega/2^N),$$

where $M_0(\omega) = H_0(\omega)/\sqrt{2}$. Using $\hat{\varphi}(0) = \int \varphi(t) dt = 1$ as $N \to \infty$ gives the famous infinite product

$$\hat{\varphi}(\omega) = \prod_j^{\infty} M_0(\omega/2^j). \tag{2.57}$$

For (2.57) to converge, it is necessary that [117]

$$H_0(\omega = \pi) = 0, \tag{2.58}$$

or in time-domain

$$\sum_k h_0(2k) - \sum_k h_0(2k+1) = 0.$$

Using (2.58) in (2.42) produces another useful requirement on $H_0(\omega)$:

$$H_0(\omega = 0) = \sqrt{2}. \tag{2.59}$$

More generally, $H_0(\omega)$ must have at least N+1 zeros at $\omega = \pi$ to achieve regularity of order N. It should be noted however that the regularity order N is not guaranteed,

as was shown in Daubechies' family of maxflat filters which have maximum zeros at $\omega = \pi$ but are not maximally regular [24]. A more accurate method of estimating regularity is to compute the Hölder regularity using the technique from Rioul's work [90].

Because $h_0(n)$ must has at least one zero at π to converge to a continuous scaling function, the corresponding highpass filter $h_1(n)$ has at least one zero at $\omega = 0$. Since $\hat{\varphi}(0) = 1$, it follows that $\hat{\upsilon}(0) = \int \upsilon(t)dt = 0$. Using the moment theorem of the Fourier transform, we have

$$\int_{-\infty}^{\infty} t^n \upsilon(t)dt = 0, \quad n \le N, \tag{2.60}$$

That is, the first N (vanishing) moments of the wavelet are zero. In general, if $h_0(n)$ has an $(N+1)$th-order zero at $\omega = \pi$, the wavelet has N zero moment. Therefore, a function with more vanishing moments is expected to have higher regularity, and vice versa.

Now, the key question is: How much regularity is useful in image coding applications? When a signal is represented in terms of transform coefficients, the signal is described as a superposition of basis functions weighted by the transform coefficients. As a result, a smooth signal can be represented by a few transform coefficients if the basis functions are smooth. Therefore, intuitively, filters with more regularity should be helpful (since most images have dominant low frequency content). However, there has been no quantification of the optimum regularity of a wavelet for coding applications. Rioul [89] compared the compression gain for orthonormal filters of varying regularity used in wavelet coders and concluded that some regularity is desired (the performance with no regularity is poor) and higher regularity helps, but not substantially. Villasenor *et al.* [114] investigated the regularity criterion of linear-phase filters and also came to the same conclusion that the performance with no regularity (Haar filter) is poor compared to those with 1 or 2 regularity, but the performance with higher regularity is worse than the Haar filter! In summary, regularity alone is not a sufficient criterion for filter choice, and as demonstrated by Balasingham and Ramstad [14], by enforcing only 2 zeros at $\omega =$

π (maximum regularity of order 1 possible) and using the extra freedom to optimize the coding gain, significant improvement in performance is possible.

2.5 Wavelet Packets

We mentioned in Section 2.3 that wavelet transform is designed to have good time resolution and poor frequency resolution at high frequencies, and good frequency resolution and poor time resolution at low frequencies. From a coding point of view, this approach is ideal for stationary signals, or signals with dominant low frequency content since most of the energy will be captured into a few coefficients. However, if this is not the case, then wavelet transform may not work well. A better alternative is the idea of wavelet packets, or arbitrarily subbands coding trees. Wavelet packets (WP) represent a generalization of the method of multiresolution decomposition, and comprise the entire family of subband-coded decompositions, including the well-known wavelet basis and the Short-Time-Fourier-Transform (STFT)-like basis as its member. Figure 2.2 shows some permissible depth-3 decompositions. A more informative diagram can be found in Figure 5.1 of Chapter 5 where the time-frequency tilings (Heisenberg box plot) from expansions using different bases are shown. The most attractive feature of WP is that they offer fast access to a rich menu of orthonormal or biorthogonal (depending on the filter set used) bases, from which the "best basis" (depending on some coding criterion) can be chosen. This enables a WP coder to exhibit, for example, a STFT-like character-

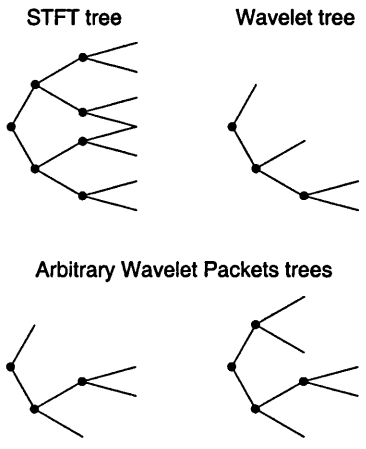


Figure 2.2 Some permissible depth-3 decompositions.

istic at one source instance, a wavelet characteristic at another instance, or any intermediate characteristic at yet other instances, to best match the signal's nonstationary statistics.

To illustrate the coding efficiency of WP, PSNR results from a wavelet transform coder and a wavelet packets coder on two (perhaps the most referenced) images are shown in Figure 2.3. The wavelet transform coder was iterated to a maximum tree depth of 6 and bit allocation was performed so that MSE is minimized, i.e. one scalar quantizer per subband. The wavelet packets coder partitioned the image into 16 sub-images and a maximum depth of 4 was used on each sub-image. First-order entropy was used as the split criterion at each decision node in

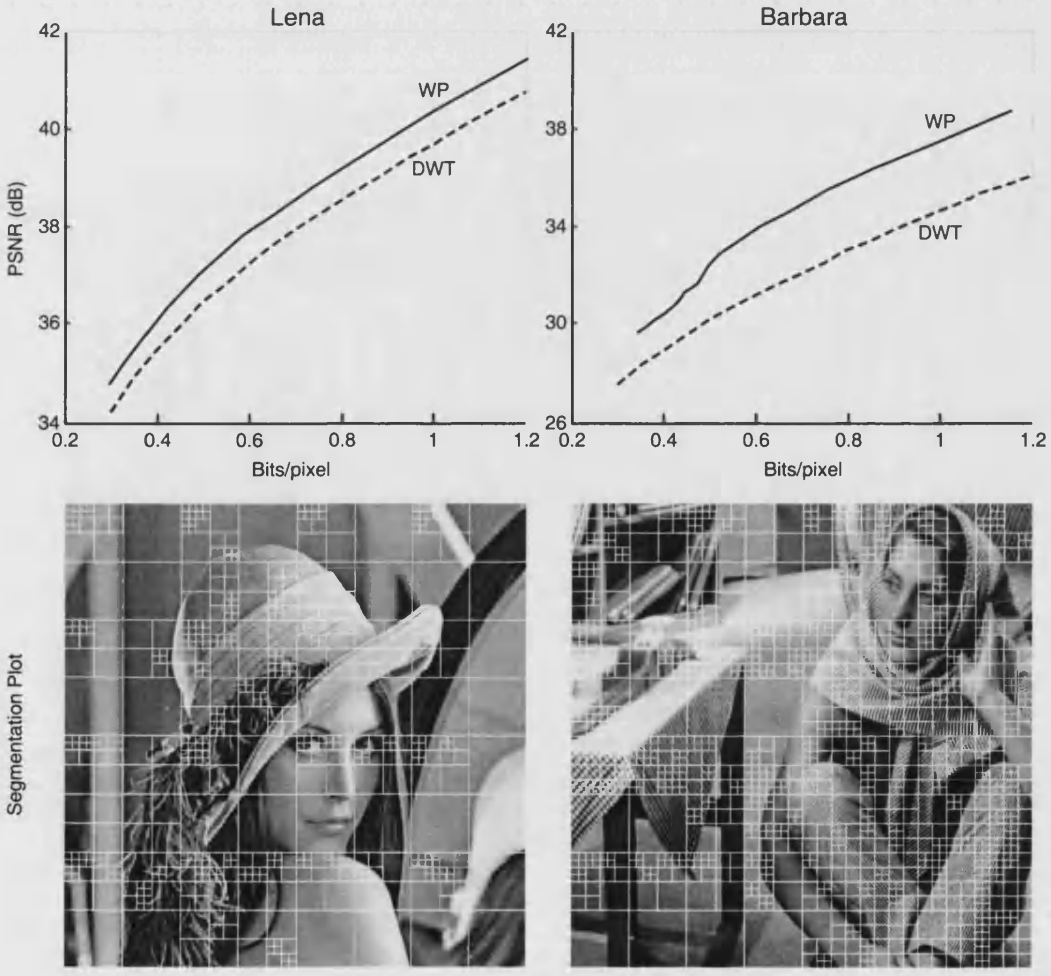


Figure 2.3 PSNR comparison between two compression systems using wavelet transform and wavelet packets.

the quadtree segmentation of each sub-image, and only one scalar quantizer was used throughout the image. As can be seen from the figure, the WP coder performs substantially better than the DWT coder by an average of 1.0 dB for *Lena* and 2.0 dB for *Barbara* in all bit rates considered.

2.6 Finite-Length Signals

Most of the literature on filter bank theory has assumed the signal to be of infinite length. This model is very reasonable if the data stream is audio signal or speech. However, in other applications like image processing, we have a signal of definite length L. Thus the finiteness of the signal $x(0),x(1),\ldots,x(L\text{-}1)$ must be taken into account. The immediate problem is filtering a finite signal. The computation of $\sum h(n)x(k\text{-}n)$ may ask for $x(\text{-}1)$, which is not defined. The most obvious approach is to view a finite length signal as being part of an infinite signal, which is zero elsewhere (known as zero padding). Although this solves the above-mentioned problem, it introduces artificial high-frequency components, which can reduce the coding gain, and redundancy in the subband signals. If the input data sequences $x(n)$ is of length L, after it is processed by the analysis filters of length N, the subband signals will be of length $N+L\text{-}1$. Including the downsampling process, there are more data points in the subbands than the input. It is therefore obvious that a better method is required, not only to define the signal beyond the boundary, but also to solve the problem of redundancy and discontinuity at the boundary. Several methods have since been developed and they are *Boundary value replication*, *Circulant extension* (circular convolution), *Symmetric extension* [15] and *Boundary filters* [36].

Boundary value replication is similar to zero padding, but instead of assuming zeros beyond the boundary, the first and last sample value is assumed to repeat indefinitely. This eliminates the artificial discontinuity at the boundaries, but the redundancy remains. Although we can discard coefficients at the boundaries to make the system non-expansive, it introduces distortion at the boundaries even

when the filter banks in use are perfect reconstruction. In the circulant extension method, the finite length signal is made periodic and is given by

$$\tilde{x}(n) = x(n \bmod L). \tag{2.61}$$

Here distortion and redundancy are eliminated, since the subband signals are $L/2$ periodic, only L points are required to transmit to the synthesis section. However, there is still an artificial discontinuity at the boundary, since the data is "wrapped around", and the beginning and end are filtered as though they are physically contiguous.

Symmetric extension again processes a periodic version of the signal, but now of period $2L$, where there is no discontinuity at either boundary. It has been shown in [15] that if linear-phase filters are used, then perfect reconstruction without distortion or redundancy or any high-frequency components is possible. Two examples of signal extension for two pairs of symmetric filters are shown in Figure 2.4. The key to non-expansive filtering is that the missing samples at the boundaries, which are required for the inverse operation to achieve perfect reconstruction, can be found within the retained samples.

The method of boundary filter [36] applies time varying filtering to the finite-length signal directly, without recourse to extension of the data. The idea is

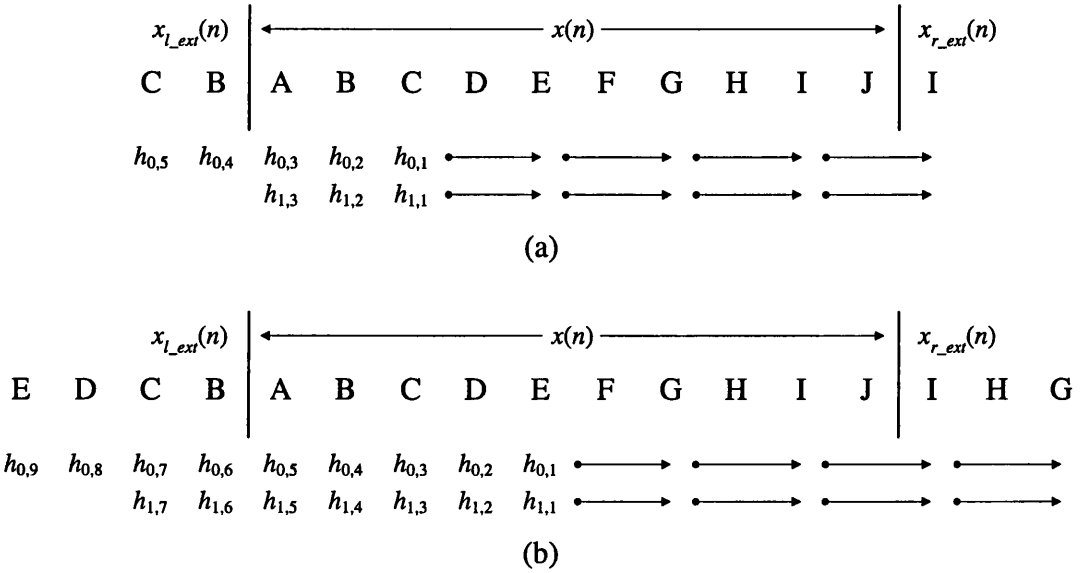


Figure 2.4 Symmetric extension for the 5/3 (a) and 9/7 filters sets (b).

A	B	C	D	E	F	G	H	I	J
$l_{0,4}$	$l_{0,3}$	$l_{0,2}$	$l_{0,1}$						
$l_{1,4}$	$l_{1,3}$	$l_{1,2}$	$l_{1,1}$						
$h_{0,6}$	$h_{0,5}$	$h_{0,4}$	$h_{0,3}$	$h_{0,2}$	$h_{0,1}$				
$h_{1,6}$	$h_{1,5}$	$h_{1,4}$	$h_{1,3}$	$h_{1,2}$	$h_{1,1}$				
		$h_{0,6}$	$h_{0,5}$	$h_{0,4}$	$h_{0,3}$	$h_{0,2}$	$h_{0,1}$		
		$h_{1,6}$	$h_{1,5}$	$h_{1,4}$	$h_{1,3}$	$h_{1,2}$	$h_{1,1}$		
				$h_{0,6}$	$h_{0,5}$	$h_{0,4}$	$h_{0,3}$	$h_{0,2}$	$h_{0,1}$
				$h_{1,6}$	$h_{1,5}$	$h_{1,4}$	$h_{1,3}$	$h_{1,2}$	$h_{1,1}$
						$r_{0,4}$	$r_{0,3}$	$r_{0,2}$	$r_{0,1}$
						$r_{1,4}$	$r_{1,3}$	$r_{1,2}$	$r_{1,1}$

Figure 2.5 Filtering operation using Daubechies D_6 filter and its boundary filters.

Table 2.2 Coefficients of the boundary filters for Daubechies D_4, D_6 and D_8 filters set (highpass filters not optimized for DC leakage).

Daubechies Filter	Boundary filters ($l_n \rightarrow$ left, $r_n \rightarrow$ right)
D_4	l_0 = [0.8660 0.5000] r_0 = [-0.5000 0.8660]
D_6	l_0 = [-0.1356 0.4619 0.8104 0.3341] l_1 = [0.9326 0.3582 -0.0411 -0.0170] r_0 = [0.0722 -0.1750 -0.2766 0.9421] r_1 = [0.3267 -0.7923 0.5149 -0.0211]
D_8	l_0 = [-0.9707 -0.1601 0.1707 0.0550 0 0] l_1 = [0.0599 -0.1824 -0.0331 0.6299 0.7156 0.2306] l_2 = [-0.0315 0.6550 -0.7195 -0.2288 0.0034 0.0011] r_0 = [-0.0161 0.0498 0.0467 -0.2833 -0.0424 0.9555] r_1 = [0.2301 -0.7139 0.6336 0.0169 -0.1890 0.0067] r_2 = [0 0 0.2203 -0.6836 0.6711 -0.1837]

that at the beginning and end of the data, a special set of boundary filters is applied, but away from the boundary, filtering is done in the usual way. Figure 2.5 illustrates the filtering operation using the D_6 filter and its boundary filters, which are listed in Table 2.2. Also included in the table are the boundary filters for D_4 and D_8.

Note that the subscript at the extreme right of each variable in Figure 2.5 indexes the filter coefficients. The construction of boundary filters is, in fact, almost simple enough to do by hand. In what follows, we will focus on orthogonal filter banks (since for the popular biorthogonal filter banks, symmetric extension is often used). One convenient way to look at multirate filter banks is in terms of the time domain operator notation. Here the analysis matrices begin with the lowpass $h_0(n)$ and the highpass $h_1(n)$. The rows have a double shift from the downsampling operation and form the doubly infinite block Toeplitz matrix

$$\mathbf{T} = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & h_0(N) & h_0(N-1) & h_0(N-2) & \cdots & h_0(0) & 0 & 0 & \cdots \\ \cdots & h_1(N) & h_1(N-1) & h_1(N-2) & \cdots & h_1(0) & 0 & 0 & \cdots \\ \cdots & 0 & 0 & h_0(N) & \cdots & h_0(2) & h_0(1) & h_0(0) & \cdots \\ \cdots & 0 & 0 & h_1(N) & \cdots & h_1(2) & h_1(1) & h_1(0) & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

It is readily verifiable that if these filters form an orthogonal filter set then

$$\mathbf{T}\mathbf{T}^T = \mathbf{T}^T\mathbf{T} = \mathbf{I}.$$

When signals have finite length $L > N$, the infinite matrix $\mathbf{T}$ must change to $L \times L$. The "middle" of the matrices is not affected, but the "ends" will be new. The goal then is to choose those end rows – the boundary filters – such that the orthogonality condition holds. Note that since we want to retain orthogonality, the truncated matrix must be square. Now consider a signal with $L = 8$, filtered by the D_4 filter set with coefficients (a, b, c, d). Two possibilities for the analysis matrix are

$$\mathbf{T} = \begin{bmatrix} -c & d & & & & & & \\ d & c & b & a & & & & \\ -a & b & -c & d & & & & \\ & & d & c & b & a & & \\ & & -a & b & -c & d & & \\ & & & & d & c & b & a \\ & & & & -a & b & -c & d \\ & & & & & & d & c \end{bmatrix} \text{ or } \begin{bmatrix} c & d & a & & & & & \\ b & -c & d & & & & & \\ & d & c & b & a & & & \\ & -a & b & -c & d & & & \\ & & & d & c & b & a & \\ & & & -a & b & -c & d & \\ & & & & & d & c & b \\ & & & & & -a & b & -c \end{bmatrix}. \tag{2.62}$$

Both choices are still of full rank, i.e. the rows containing the truncated filters are linearly independent of each other, and of other rows. The truncated matrix is no

longer unitary since the rows containing the truncated filters are not orthogonal. However, because we have a full set of independent vectors, we can restore orthogonality using the Gram-Schmidt procedure. Let $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ denote the top rows containing the truncated filters. The orthonormal vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ for the span of $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ are computed as

$$\mathbf{u}_j = \frac{\mathbf{w}_j}{\|\mathbf{w}_j\|}, \quad j = 1, 2, \ldots, k, \tag{2.63}$$

where

$$\mathbf{w}_j = \mathbf{v}_j - \sum_{i=1}^{j-1} \langle \mathbf{v}_j, \mathbf{u}_i \rangle \mathbf{u}_i. \tag{2.64}$$

The same procedure is then used on the end rows. Solving the truncated matrix on the left of (2.62) yields the coefficients listed in Table 2.2. For the matrix on the right, the coefficients for the boundary filters are

$$\begin{aligned} l_0 &= [\ 0.2260 \quad 0.8436 \quad 0.4871] \\ l_1 &= [-0.9741 \quad 0.1958 \quad 0.1130] \\ r_0 &= [-0.1478 \quad 0.2560 \quad 0.9553] \\ r_1 &= [\ 0.4777 \quad -0.8273 \quad 0.2956]. \end{aligned} \tag{2.65}$$

Obviously, the boundary filter is not unique, i.e., there exist more than one solution for the D_4 filter. Moreover, in the above two examples, there are, in fact, countable infinite solutions, since a different set of linearly independently vectors as the input to the Gram-Schmidt procedure produces a different set of orthogonal boundary vectors. To explore the whole space of possible solutions, we can pre-multiply the first two rows of the matrix on the right of (2.62) by any 2×2 orthogonal matrix, e.g. Givens rotation. We can then use that freedom to make the truncated filter on the second row a highpass, i.e. $l_1(0) + l_1(1) + l_1(2) = 0$. Similarly $r_1(0) + r_1(1) + r_1(2) = 0$ makes the last row orthogonal to DC input. Note that strict enforcement of the highpass criterion sometimes causes the corresponding lowpass to have very bad high frequencies suppression. In that case, equating the sum to an arbitrary small value usually helps. However, it must be emphasized that in coding applications where the highpass signal is often quantized more heavily than those in the lowpass, it is more important for the highpass to have no DC leakage rather than for the lowpass to have good frequency response. Figure 2.6 shows the frequency re-

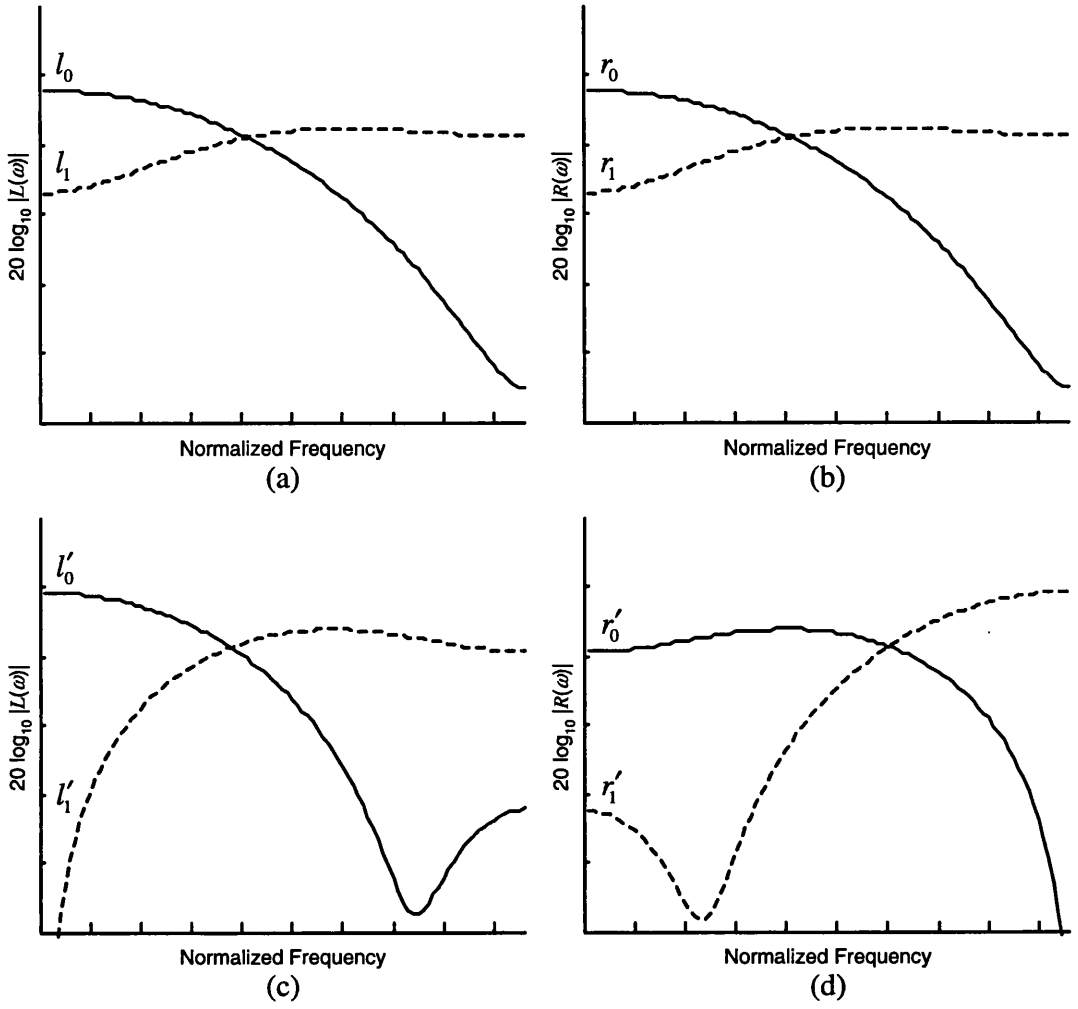


Figure 2.6 Frequency response of D_4 boundary filters before and after optimization. (a) Left and (b) right boundary filters before optimization. (c) Left and (d) right boundary filters after optimization.

sponses of the boundary filters in (2.65), as well as its optimized version. In Figure 2.7, the frequency responses of the D_6's boundary filters are shown. Notice that in the latter case, only the optimized filters possess the desired frequency responses.

Of the few methods discussed, it is apparent that both symmetric extension and boundary filters achieve the desired goal. In practice, however, symmetric extension have been preferred in almost all wavelet compression systems because of its simplicity and the fact that "good" symmetric filters are abundant in the literature today. Unless stated otherwise, all results reported in this thesis use biorthogonal filters, and hence symmetric extension at the signal boundary.

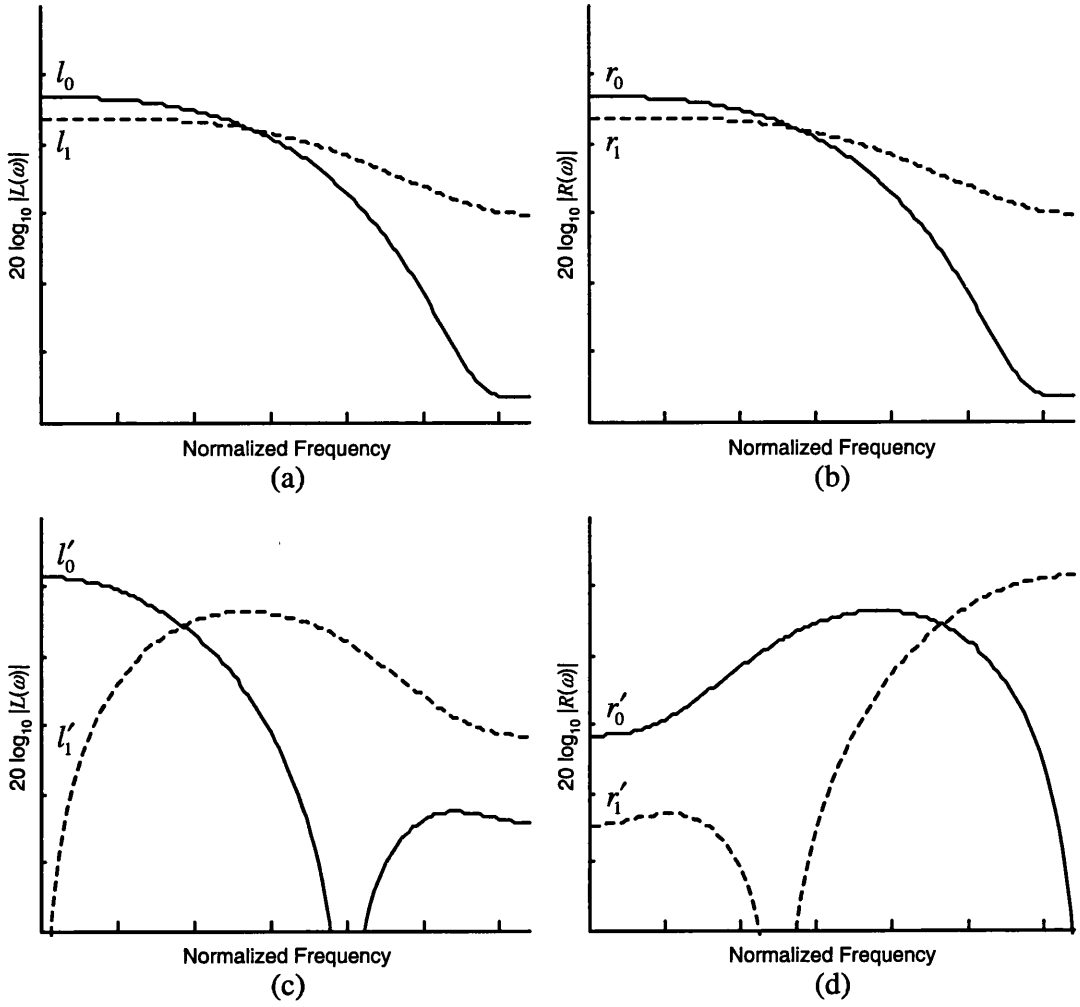


Figure 2.7 Frequency response of D_6 boundary filters before and after optimization. (a) Left and (b) right boundary filters before optimization. (c) Left and (d) right boundary filters after optimization.

2.7 Summary

In this chapter we have studied the necessary conditions for a perfect reconstruction 2-channel filter bank design and its connection to a more recent advance, the wavelet transform. Construction method for the famous Daubechies' family of maxflat filters was also briefly discussed. Some simple coding examples using the discrete wavelet transform (DWT) and its elegant generalization, or wavelet packets (WP), have demonstrated the advantages that the WP have over the DWT with fixed time-frequency tilings. Unlike transform coding where the finiteness of the signal poses

no problem, DWT and WP require special treatment at the signal boundary. Of the few signal extension methods discussed, symmetric extension has been preferred in almost all wavelet-based compression systems. However, its requirement for linear-phase filters has made 2-channel orthogonal filter bank design with good frequency response impossible (except Haar wavelet).

Chapter 3

Motion-Compensated Residual Pictures

3.1 Introduction

In Chapter 1, we saw that both spatial and temporal redundancy reduction techniques are needed for a high compression ratio in a video coding system to achieve practical transmission rates and storage requirements. Within the framework of the international coding standards, techniques similar to JPEG are used to reduce spatial redundancy; that is, each picture of the video is individually and independently compressed using DCT and Huffman coding. Such a method of compression is commonly referred to as an intra-coding technique. To exploit the high degree of correlation between adjacent pictures, both H.26x and MPEG employed motion estimation and compensation to form an inter-frame prediction error signal. This error signal, also known as motion residual, is then compressed using a similar DCT-based coding technique, called an inter-coding technique.

Figure 3.1 shows an example of frames from some common test sequences and their motion compensated inter-frame counterparts. The motion compensated residuals clearly look very different from the "normal" pictures on the left with which we have frequently come into contact in our daily lives. The motion residuals can be characterized, in general, as having predominantly high frequency content and fine line structures generated by the edges of the features in the video

scene. These characteristics are significantly different to those of the "normal" pictures. To quantify these differences, the power spectrum, autocorrelation function and histogram plot for the sequences shown in Figure 3.1 are depicted in Figure 3.2. It can be seen from the figure that the energy of the motion residuals is more evenly distributed across the frequency spectrum and that there is little correlation between samples in a motion compensated picture. Both observations, when taken within the context of source coding, essentially render the rationale behind using linear transformation for de-correlation and energy compaction, as is done within the standards, less relevant. This leads us to an obvious question: Can we omit DCT or any other linear transformation on the motion residuals in the first place? Also, and more importantly, will DWT provide any significant objective gain over DCT?

In this chapter, we first approach the problem from a theoretical viewpoint. Specifically, the rate-distortion bounds for motion residual with and without linear transforms are established. The results will provide valuable insight into the per-

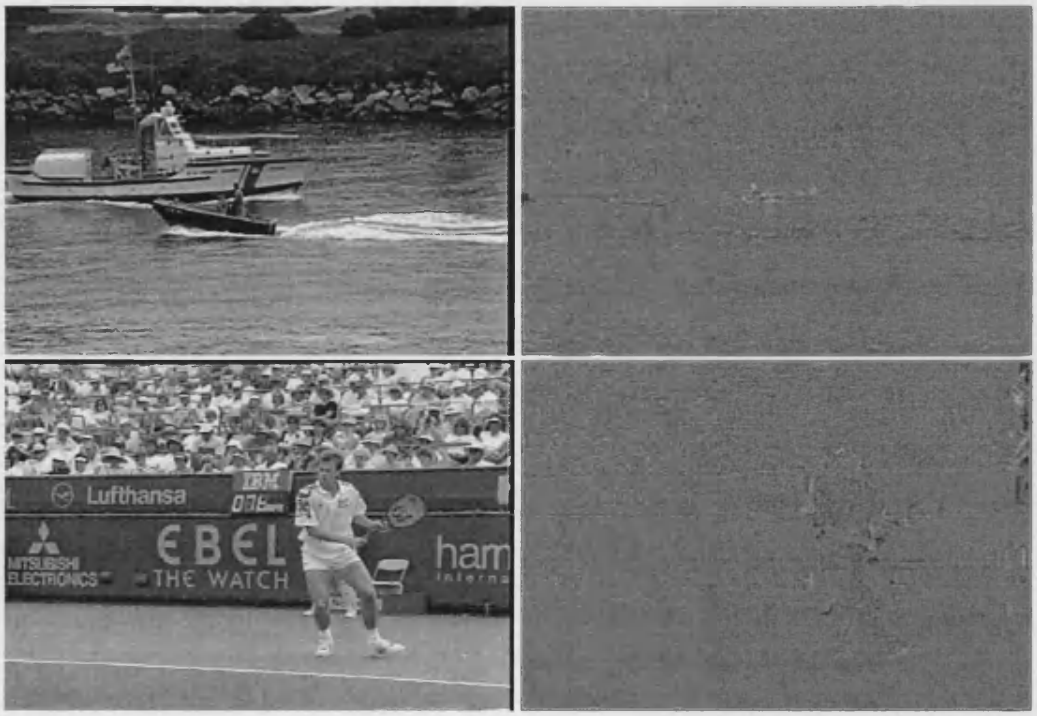


Figure 3.1 Example of a frame from *Coastguard* (top) and *Stefan* (bottom) sequence and their motion compensated inter-frame counterparts.

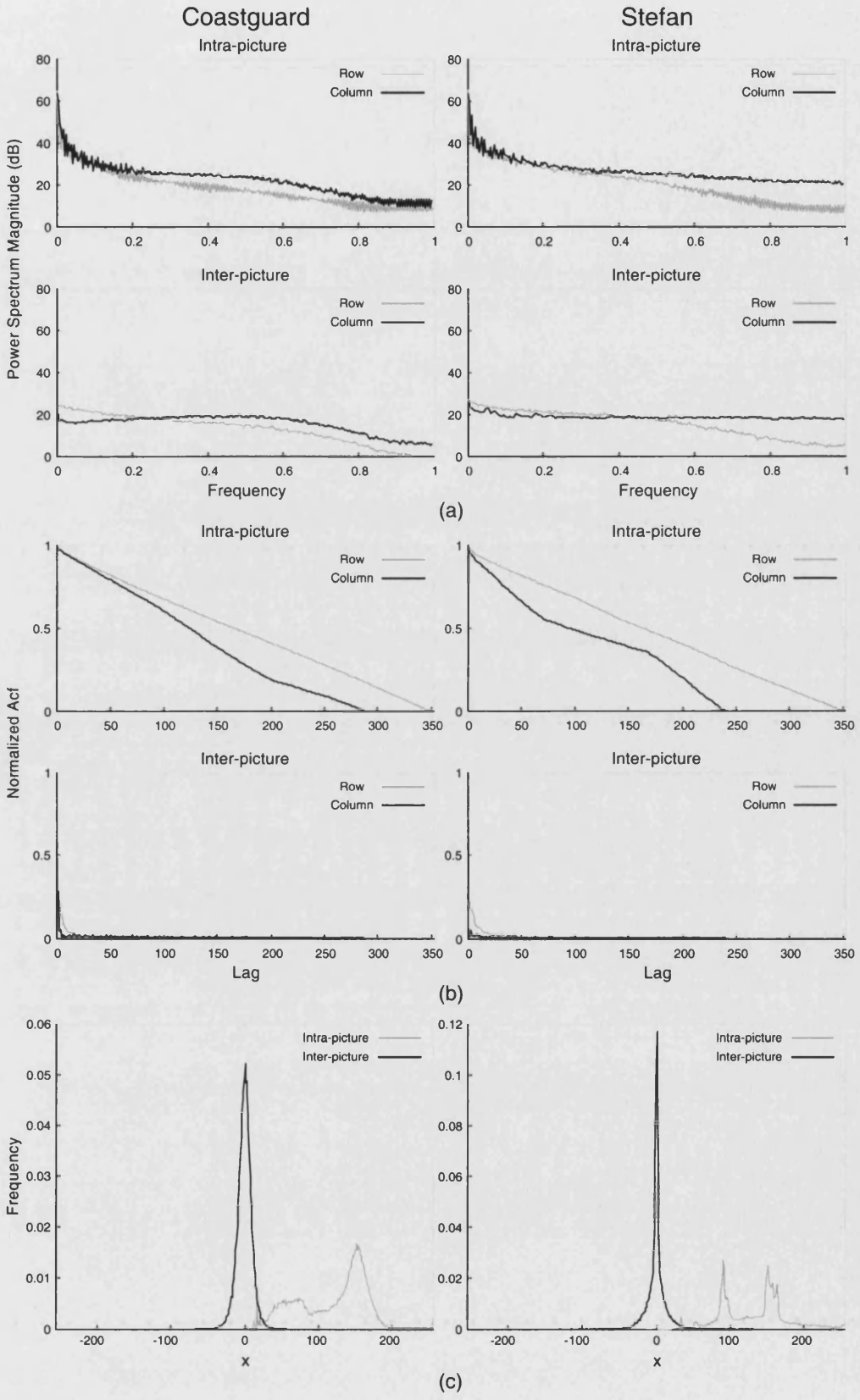


Figure 3.2 Power spectrum magnitude (a), autocorrelation function (b) and histogram plot (c) for the pictures shown in Figure 3.1.

formance of various transforms, e.g. DCT, DST and DWT, considered in this work. We then investigate the operational rate-distortion performance from a practical setting; that is, single or multiple quantizers will be designed for each particular sources. Finally, we explore the use of an adaptive tree structure using wavelet packets for motion compensated residual coding.

3.2 Rate-Distortion Function (RDF)

Rate-distortion theory is a well-known theory of lossy source coding initiated by Shannon in his celebrated papers [8, 95]. Its focus has been to a large extent the derivation of performance bounds; that is, determining the region of achievable points in the bits-fidelity trade-off for stationary ergodic sources satisfying a moment finiteness condition. The performance that it prescribes is approachable only in the limit as code dimension increases. Less well known is another asymptotic theory of lossy source coding, which goes by the names of high-rate, high-resolution and asymptotic quantization theory [29]. Unlike the Shannon theory, the high-rate quantization theory prescribes the best performance of codes with a given dimension and an asymptotically large rate. A comparison between these two approaches can be found in [79].

Since rate-distortion performance is the fundamental trade-off in the design of any lossy compression system, RDF provides a useful and intuitive starting point in addressing the usefulness of using linear transformation on the motion residuals. In particular, the rate-distortion performance bound of motion residuals with and without 2-D linear transform should provide a valuable insight into the question at hand. Unfortunately, to derive bounds one needs to first characterize the sources and this can be problematic for complex sources such as video. Nevertheless, it is well known in the coding community that marginal statistics of the prediction error frames are highly non-Gaussian. In particular, the histograms (see Figure 3.1 (c)) are found to have much heavier tails and are more sharply peaked than one would expect from a Gaussian density. Furthermore, it has already been shown in [7, 100,

134] that distribution of wavelet coefficients and DCT coefficients of motion compensated video and general images possess Laplacian distribution.

In what follows we assume that the probability density function (pdf) of the assumed memoryless source as having a zero mean Generalized Laplacian Distribution given by [100]

$$p(x)=\left[\frac{\beta}{2\alpha\Gamma(1/\beta)}\right]\exp\left\{-\left|\frac{x}{\alpha}\right|^{\beta}\right\}, \tag{3.1}$$

where $\Gamma(.)$ is the standard Gamma function. This distribution is also commonly called the exponential power distribution in statistic. Its kurtosis coefficient is

$$\gamma=\frac{\Gamma(5/\beta)\Gamma(1/\beta)}{\Gamma(3/\beta)^2}. \tag{3.2}$$

For $\alpha=\sqrt{2}$ and $\beta=2$, (3.1) reduces back to a standard Gaussian distribution with a kurtosis coefficient of three. Since the maximum likelihood method is statistically efficient at obtaining parameters estimates, we will focus on it in what follows. The likelihood of x is

$$L(\alpha,\beta|x)=\prod_{i}^{N}p(x_i) \tag{3.3}$$

and the log likelihood is given by

$$\begin{aligned}\log L(\alpha,\beta|x)&=\log\prod_{i}^{N}p(x_i)\\&=\log\prod_{i=1}^{N}\frac{\beta\exp\{-|x_i/\alpha|^{\beta}\}}{2\alpha\Gamma(1/\beta)}\\&=\sum_{i}^{N}\left[\log\beta-\log 2-\log\alpha-\log\Gamma(1/\beta)-|x_i/\alpha|^{\beta}\right].\end{aligned} \tag{3.4}$$

To find $\{\alpha,\beta\}$ which maximize this expression, we take the derivative with respect to $\{\alpha,\beta\}$ and set it to zero. One then has

$$\alpha=\left[\frac{\beta}{N}\sum_{i}x_i^{\beta}\right]^{1/\beta} \tag{3.5}$$

$$\frac{N}{\beta}+\frac{N}{\beta^2}\frac{\Gamma'(1/\beta)}{\Gamma(1/\beta)}=\sum_{i}(x_i/\alpha)^{\beta}\log(x_i/\alpha) \tag{3.6}$$

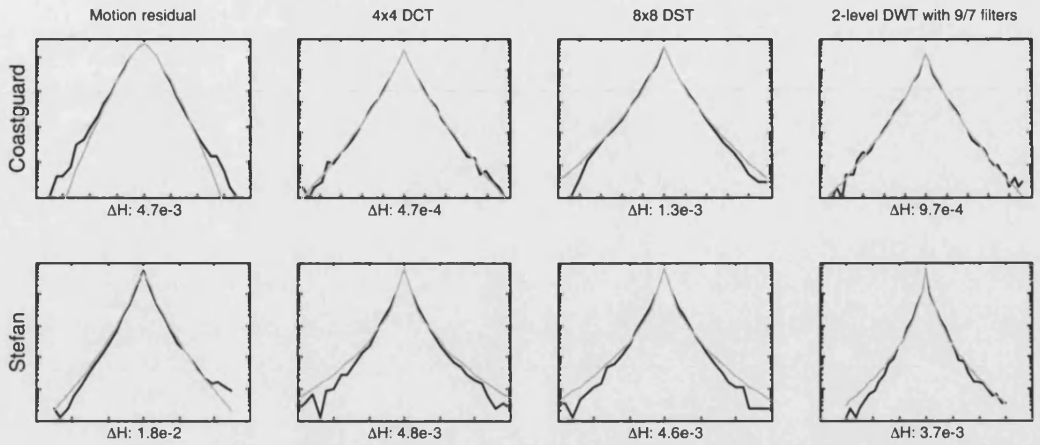


Figure 3.3 Examples of coefficient histograms (black lines) for various linear transforms and the motion residuals. Also shown (grey lines) are fitted densities corresponding to (3.5) and (3.6). Below each histogram is the relative entropy.

that are readily solved using any optimization routine. Figure 3.3 shows the histograms and the approximations of a frame from *Coastguard* and *Stefan* sequence with and without transformation. Notice that not all coefficients that are used in the estimation come from orthogonal transforms; without the orthogonality property, the performance bounds computed for that particular density distribution may not be a true indication of the efficiency of the corresponding transform, even if the model provides a perfect match.

From Figure 3.3, one can clearly see that *Stefan* sequence has the worst fit (by visual inspection) among the two images. Nevertheless, the density model fits the histograms reasonably well, as indicated by the Kullback-Leibler divergence measures or relative entropy given below each plot. The relative entropy is a measure of how different two probability distributions are and is given by

$$D(p\|q) = \sum_i p(x_i) \log\left(\frac{p(x_i)}{q(x_i)}\right). \tag{3.7}$$

The relative entropy between $p(x_i)$ and $q(x_i)$ can also be seen as the average number of bits that are wasted by encoding events from a distribution p with codes based on a "not-quite-right" distribution q.

Having determined the best estimates through (3.5) and (3.6), the rate-distortion performance bounds for the various density distributions is readily computed using the Blahut algorithm [12]. Table 3.1 lists the signal-to-quantization-noise ratio

$$QSNR = 10\log_{10}\left(\sigma_x^2/MSE\right) \tag{3.8}$$

of various density distributions with parameters estimated from the transform coefficients that can be achieved by an ideal multi-dimensional quantization scheme, together with the relative entropy of the actual histogram and its approximation. Note that several different implementations of DWT indicated by the name "DWT xx-y" have been shown in the table. "xx" refers to the kernel used while "y" is the number of iterations that DWT has been performed in a dyadic structure; for example, "97-2" means that the famous bi-orthogonal 9/7 filters have been used and the

Table 3.1 Comparison of QSNR performance of various Generalized Laplacian distributions with estimated parameters.

Coastguard

BPP	Residual	DCT 4x4	DCT 8x8	DST 8x8	DWT 97-1	DWT 97-2	DWT 97-3	DWT 22-1	DWT 22-2	DWT 22-3
0.0625	0.453	0.798	**1.022**	0.992	0.773	**0.863**	0.836	0.566	0.597	0.603
0.125	0.881	1.441	**1.753**	1.721	1.394	**1.505**	1.488	1.066	1.120	1.128
0.25	1.688	2.505	**2.932**	2.902	2.442	**2.565**	2.577	1.963	2.039	2.047
0.5	3.235	4.340	**4.897**	4.866	4.250	**4.398**	4.427	3.600	3.708	3.709
1	6.268	7.590	**8.305**	8.289	7.496	**7.660**	7.703	6.689	6.819	6.829
ΔH	4.7e-3	4.7e-4	2.2e-3	1.3e-3	1.5e-3	9.7e-4	8.0e-4	1.6e-3	2.1e-3	4.1e-3

Stefan

BPP	Residual	DCT 4x4	DCT 8x8	DST 8x8	DWT 97-1	DWT 97-2	DWT 97-3	DWT 22-1	DWT 22-2	DWT 22-3
0.0625	0.860	**1.731**	1.598	1.627	**1.691**	1.550	1.606	1.260	1.336	1.336
0.125	1.499	**2.739**	2.573	2.565	**2.685**	2.528	2.579	2.103	2.191	2.189
0.25	2.576	**4.254**	4.061	3.999	**4.171**	4.015	4.056	3.426	3.546	3.523
0.5	4.420	**6.622**	6.389	6.291	**6.476**	6.341	6.388	5.577	5.714	5.670
1	7.690	**10.44**	10.20	10.00	**10.23**	10.13	10.18	9.16	9.32	9.25
ΔH	1.8e-2	4.8e-3	7.8e-3	4.6e-3	2.4e-3	3.7e-3	2.8e-3	1.1e-2	5.0e-3	1.0e-2

lowpass subband iterated once. It is also worth noting that all motion residuals used throughout the chapter are obtained with the *Overlapped Block Matching Compensation* (OBMC) algorithm rather than the much simpler block matching technique. This is done so that we have a much "smoother" motion residual that works better with DWT.

The results of Table 3.1 clearly demonstrate that a higher signal-to-quantization ratio can be achieved with density distributions from the transform coefficients than that of the motion residuals signal. Also, of the various different implementations of DWT considered in the experiment, it is interesting to note that DWT with 3-levels of decomposition does not provide any significant performance gain over those with 1-level of decomposition. This observation is in sharp contrast to many general image wavelet coders, where several level of decompositions are usually required to achieve good coding performance.

Although the above discussion enables us to determine the boundaries between achievable and non-achievable regions, the bounds computed may not be tight for situations of practical relevance (e.g. relatively low rate and small block size). Moreover, modeling of coefficients with an i.i.d. Laplacian or Gaussian model inevitably raises the obvious question about how accurate these models might be. For example, how can one be sure that the improvement in QSNR is not due to modeling mismatches? While statistical tests like chi-square can be use to test the null hypothesis that two different data sets are drawn from the same distribution function, it is very difficult to find a model or several models that pass the test in all instances. Also, there is often considerable arbitrariness as to how the bins should be chosen during the calculation of the test statistic. Therefore, instead of trying to find or construct a better model for the problem, we assess the performance of coding motion residuals with and without linear transformation, as well as the efficiency of various transforms in a more practical setting in the next section.

3.3 Optimum Scalar Quantization Results

The scalar quantizer has often been preferred over other quantization schemes in the design of lossy compression systems because of its simplicity and good performance. One very good example is the H.26x/MPEG standard where more than one quantizer can be used for each individual picture. In this section, two bit rate control techniques that are found within the H.26x/MPEG standard will be used as the basic framework to assess the performance of motion residual coding with and without linear transformation. The first technique uses only one quantizer per frame and is known as the frame level control (FLC) while the second technique allows one quantizer per macroblock and is therefore known as the macroblock level control (MLC). Unless stated otherwise, both first order entropy and an MSE distortion criterion will be used throughout the experiments.

In our simulation of a FLC system, optimum scalar quantizers will be used for both the transform coefficients and motion residual to ensure fair comparison. Before proceeding to analyze the experimental results, it is helpful to briefly review some properties of an optimum quantizer and the method of constructing one. In the discussions that follow, only the basic ideas are presented. Many issues are necessarily treated too briefly, and the reader will need to refer to the original literature [27, 63, 67, 106, 125] for more details.

The design of optimum scalar quantizers has been widely studied for some general statistical source classes that can be used to model numerous real-life phenomena. The input and output of a quantizer to be optimized can be defined by the relationship

$$y = y_k, \text{ if } x_k < x \leq x_{k+1}, \quad k \in L, \tag{3.9}$$

where x_k and y_k are called the decision levels and reconstruction thresholds respectively. The set L of levels may either be finite ($L = \{1,2,\ldots,N\}$ for an N-level quantizer) or countable infinite. An optimum quantizer is one that minimizes the mean squared error

$$D = E\left[(x-y)^2\right] = \sum_{k}^{N} \int_{x_k}^{x_{k+1}} (x-y_k)^2 p(x)dx. \tag{3.10}$$

The necessary conditions for optimality is obtained by differentiating D with respect to the x_k's and y_k's and setting the derivatives equal to zero. One then has

$$x_k = \frac{y_k + y_{k-1}}{2}, \quad k = 2,...,N, \tag{3.11}$$

$$y_k = \frac{\int_{x_k}^{x_{k+1}} xp(x)dx}{\int_{x_k}^{x_{k+1}} p(x)dx}. \tag{3.12}$$

Both (3.11) and (3.12) are known as the *Nearest Neighbor Condition* and the *Centroid Condition* respectively. The quantizer that satisfies these conditions is called the Lloyd-Max quantizer (LMQ) [63, 67] and the well-known Lloyd-Max iterative algorithm can be found in Table 3.2. LMQ, however, will not produce optimum performance when the quantization indices are entropy coded, as is often done in many compression systems. An optimized quantizer with the entropy constraint is better suited to such a system.

Berger [9, 10] considered entropy-constrained scalar quantizers (ECSQ), and noted that the necessary conditions for ECSQ were fulfilled by certain quantizers having an infinite number of levels and equal step widths, termed uniform-threshold quantizers (UTQ). Such quantizers also satisfy the Centroid Condition and are in fact optimum for sources having an exponential pdf but not the Laplacian source, as they fulfilled only necessary but not sufficient conditions. Several design

Table 3.2 Llyod-Max Algorithm

1. Choose an initial set of reconstruction thresholds: $\{y_1, y_2, \ldots, y_N\}$
2. Update the decision levels x_k, $1 \le k \le N-1$ using (4.11).
3. Update the reconstruction thresholds using (4.12).
4. Repeat step 2 & 3 until convergence occurs:

$$|y_k - \tilde{y}_k| < \varepsilon, \quad \forall k$$

methods [27, 106] for the Laplacian source have been proposed, but in this work, we will only consider the work by Sullivan [106]. Sullivan used the memoryless property of the exponential distribution to develop a non-iterative algorithm for obtaining the optimum quantizer design. He noted that, for a squared-error distortion measure, the optimum quantizer for the Laplacian source has a dead-zone (center step) that is always larger than the size of other step. Define the dead-zone ratio z such that

$$z = t/(\alpha - \delta(\alpha)), \tag{3.13}$$

where α is the step size for the two infinite-level subquantizers to the left and right of the center dead-zone step, t is the center dead zone decision level and $\delta(\alpha)$ is the reconstruction threshold. He found that the optimum dead-zone ratio z^* always lies in the range of $0.95 < z^* < 1$. A suboptimum quantizer (with $z = 1$), termed a uniform reconstruction with unity ratio quantizer (URQ), that preserves the optimality of the reconstruction values has also been investigated in his work. URQ has a very simple decoding rule of just multiplying the quantization index by the quantizer step size. The encoding rule is slightly more involved: the right and left center decision levels are constructed as $t = t_r = t_l = \alpha\text{-}\delta(\alpha)$, while the right and left subquantizers have a uniform step size of α. $\delta(\alpha)$ is the only free parameter that has to be solved for the input distribution. Compared to a truly optimum ECSQ, URQ stays within about 0.0022 dB of optimality at all bit rates. Table 3.3 lists the QSNR per-

Table 3.3 QSNR for RDF and various scalar quantizers.

Rate (bps)	RDF	URQ	UTQ	UQ	UQDZ
0.0625	0.5571	0.5492	0.5494	0.3010	0.5122
0.1250	1.0454	0.9747	0.9741	0.5845	0.9220
0.2500	1.9191	1.7345	1.7287	1.1536	1.6591
0.5000	3.5349	3.1240	3.1046	2.3245	3.0160
1.0000	6.6117	5.8155	5.7685	4.9761	5.6215
2.0000	12.6588	11.3665	11.3320	10.9508	10.8653
4.0000	24.9619	23.1774	23.1903	23.1532	22.3997

formance of the various quantizers discussed so far. Also shown in the table are the rate-distortion function (RDF), a uniform quantizer (UQ), which can be viewed as a UTQ with suboptimum offset, and a UQ with a dead-zone twice the size of other steps (UQDZ) that approximates the operation of a bit-plane coder. It can be seen from the table, perhaps surprisingly, that UQDZ is actually very competitive in the range below 1 bits per sample (bps). Also, it is worth noting that in the context of coding applications, the parameters that characterize the quantizer that is being used in the encoder must be communicated to the decoder. For the quantizers listed, UQ and UQDZ need only the step size to be transmitted. URQ needs both the step size and the offset, while UTQ needs the step size and N (where N is the of number of quantization steps) reconstruction thresholds.

Figure 3.4 shows the PSNR performance gap of several transform coding schemes (e.g. DCT, DST and DWT with 9/7 filters) using URQ, measured with respect to the PSNR of the motion residual. Six common test sequences, namely *Coastguard*, *Mother & Daughter*, *Salesman*, *Stefan*, *Mobile-Calendar* and *Flower-Garden*, were used. Results for each test sequence were separated into two columns – block transforms on the left and DWT on the right – for easier interpretation. For low-motion sequences such as *Coastguard*, *Mother & Daughter* and *Salesman*, DCT8 and DWT2 (low-pass iterated once) generally provide the best objective gain and DCT8 is usually better than DWT2 by 0.1 – 0.2 dB. For the remaining sequences which exhibit moderate to high motion, DCT4 and DWT1 are the better choices and DWT1 outperforms DCT4 by 0.1 – 0.2 dB. Notice that a similar conclusion can also be drawn from the results that are obtained by the modeling technique, although the PSNR comparison between transforms of different bases does not quite matched up. Note that the experimental results clearly defy the commonly held belief that DST might be a better choice than DCT in motion residual coding. Significant PSNR improvement (by DCT8 over DST8) can be seen in *Mother & Daughter* and *Salesman* while the differences between the two are modest for the remaining sequences.

As mentioned at the start of this section, both H.26x and MPEG also specify another bit rate control where one quantizer is used for each macroblock. While it

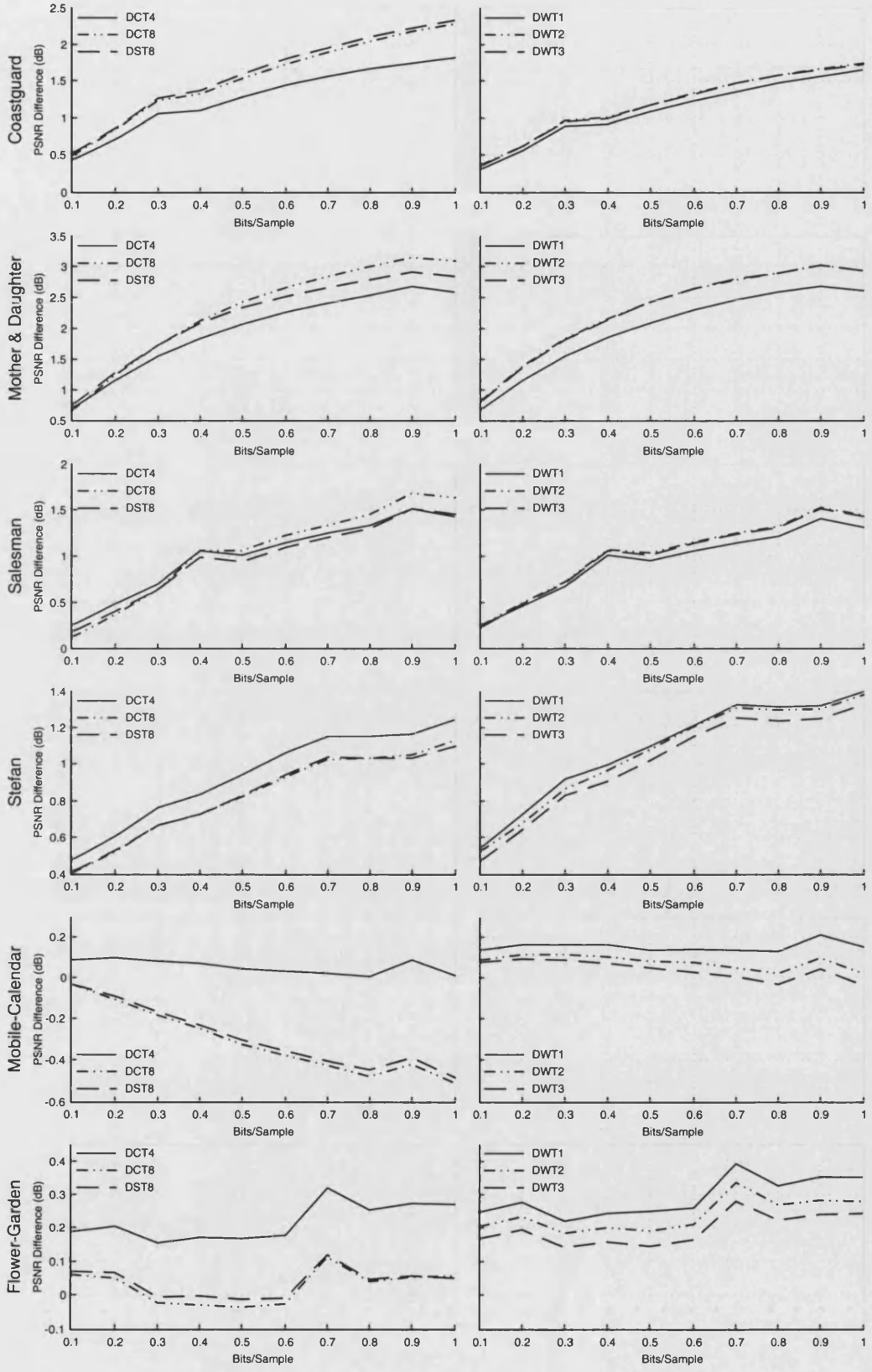


Figure 3.4 PSNR performance gap of transform coding using URQ, measured with respect to the PSNR of the motion residual.

is possible to incorporate URQ for each macroblock, a very significant portion of the available bits budget (especially for very low bit rate applications) will be taken up by the parameters that are needed to specify all the quantizers. Thus, in our simulation of such a system, the fairly competitive UQDZ will be used instead. One major problem that often arises with a multi-parameter quantization scheme is: How do we distribute the bits, or in our specific case, how do we distribute the bits among the macroblocks within each picture? In the case of a "normal" picture, the solution is often based primarily on the concept of spatial masking – the increase in threshold for a given stimulus in the presence of a masking signal. For motion residual, however, temporal masking is usually employed. The quantizer step size, for example, is often increased for areas where there is large motion. In either case, the need to consider the effect of the quantizer choice over other consecutive pictures in video applications further complicates the matter. In our current implementation, we will focus only on one single frame, as we have done in the URQ case, and the bit allocation problem is treated such that the overall MSE distortion is minimized.

Define the overall distortion D as a function of the bit allocation vector, $B = (b_1,b_2,\ldots,b_k)$ according to

$$D(B) = \sum_{i=1}^{k} W_i(b_i), \tag{3.14}$$

where $W_i(b_i)$ is the distortion function. Let S denotes the set of all admissible allocation vectors. Given a quota of bits R_c, the bit allocation problem can be formally stated as

$$\begin{aligned} &\min_{B \in S} D(B) \textit{ subject to the constraint } R(B) \le R_c, \\ &\text{where } R(B) = \textstyle\sum_{i=1}^{k} b_i. \end{aligned} \tag{3.15}$$

The study by Huang and Schultheiss [42] in their original paper on transform coding provides a simple and elegant solution to (3.15) by using the high-resolution quantization approximations so that an explicit expression for the optimum distortion function is available. For identically distributed normalized sources, the optimum bit assignment is given by

$$b_i = \bar{b} + \frac{1}{2}\log_2 \frac{\sigma_i^2}{\rho^2}, \tag{3.16}$$

where

$$\bar{b} = R_c / k$$

is the average number of bits per parameters (e.g. subbands), k is the number of parameters, and

$$\rho^2 = \left(\prod_{i=1}^{k} \sigma_i^2 \right)^{\frac{1}{k}}$$

is the geometric mean of the variances of the input sources. The minimum overall distortion attained with this solution is given by

$$D = \frac{k\rho^2 2^{-2\bar{b}}}{12} \left\{ \int_{-\infty}^{\infty} f^{\frac{1}{3}}(x)\,dx \right\}^3, \tag{3.17}$$

where $f(x)$ is the input pdf. If Gaussian source is assumed, (3.17) simplified to

$$D = \sqrt{3}k\pi\rho^2 2^{-(2\bar{b}+1)}. \tag{3.18}$$

The result in a nutshell, simply says that the number of quantization steps N_i of the ith quantizer should be proportional to the standard deviation of the random variable that it quantizes. Unfortunately, the high rate approximation from which the results were derived may not be appropriate for our very low bit rate experiments. Instead, we solve the optimization task in (3.15) by using the Lagrangian technique [81, 94, 96] that has gained popularity due to its effectiveness, conceptual simplicity, and it's ability to effectively evaluate a large number of possible coding choices in an optimized fashion.

The Lagrangian formulation of the unconstrained minimization problem in (3.15) is given by

$$\min J(\lambda), \text{ where } J(\lambda) = D(B) + \lambda R(B), \tag{3.19}$$

where the Lagrangian rate-distortion function $J(\lambda)$ is minimized for a particular value of the Lagrange multiplier λ. Each solution to (3.19) for a given value of the Lagrange multiplier λ corresponds to an optimum solution to (3.15) for a particular value of R_c. In other words, the operational rate-distortion hull can be obtained sim-

ply by swiping the Lagrange multiplier λ through all positive real numbers. For most applications where only a finite number of bit rates $R_{Target} = \{R_{c1},R_{c2},\ldots,R_{cn}\}$ are of interest, fast convex search algorithms are often used to find the optimum operating value λ^*. One such algorithm is the bisection search method, which has gained popularity due to its simplicity and great speed. Two initial values, λ_u and λ_l with $\lambda_u \geq \lambda_l$, that are required by the bisection search method must also satisfy the condition

$$\sum_i R_i^*(\lambda_u) \leq R_{Target} \leq \sum_i R_i^*(\lambda_l), \tag{3.20}$$

where $R_i^*(\lambda)$ is the rate associated with the optimum quantizer choice for the ith block. Note that failure to find any λ_u and λ_l that satisfy (3.20) means that the given problem is unsolvable, i.e. the R_{Target} is inconsistent with the given sets of quantiz-

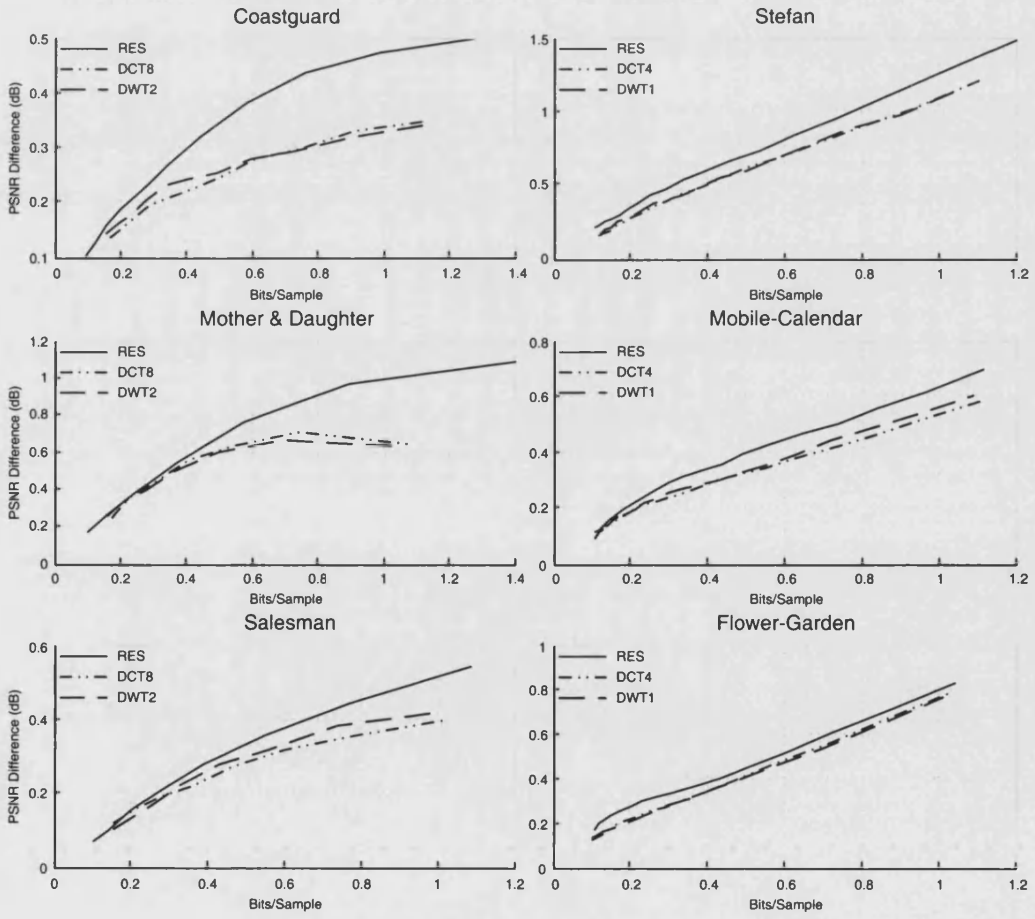


Figure 3.5 PSNR performance gap of multi-parameters quantization scheme, measured with respect to a single UQDZ version.

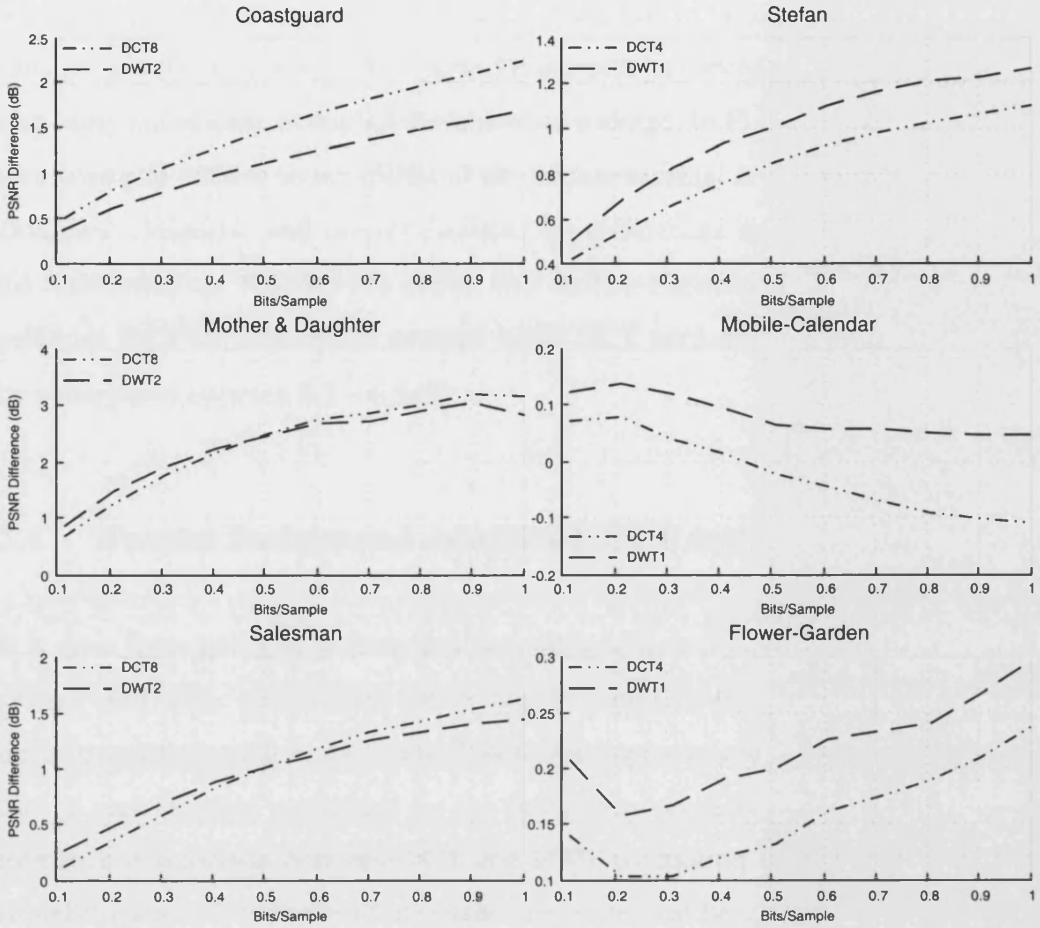


Figure 3.6 PSNR performance gap of transform coding using multiple UQDZ, measured with respect to the PSNR of the motion residual.

ers. A conservative choice for a solvable problem would be $\lambda_u = \infty$ and $\lambda_l = 0$. It must be emphasized that regardless of its implementation detail, the optimization routine requires a one-time fixed cost where the set of all $\{R,D\}$ operating points (e.g. $W_i(b)$) must be pre-calculated for the entire admissible quantizers set – a pre-requisite that might prove too costly for some applications.

Figure 3.5 shows the PSNR performance gain that can be achieved with a multi-parameters quantization scheme, measured with respect to the single parameter version. Only three results (e.g. motion residual and two selected transforms) are plotted in order to maintain the figure readability. It can be seen clearly from the figure that motion residuals without any transformation benefit the most from the use of a multiple quantizers scheme. PSNR performance gain as high as 1.0 dB

has been achieved for *Stefan* and *Mother & Daughter* at bit rates of 0.8 and 1.0 bpp (bits per pixel) respectively. The performance gain achieved by both transforms is also fairly significant, about 0.3 dB higher on average. In Figure 3.6 the PSNR performance gap relative to the PSNR of the motion residual is plotted. For *Mother & Daughter*, *Salesman* and *Flower-Garden*, the differences in performance between the transforms are modest. For *Stefan* and *Mobile-Calendar*, however, DWT outperforms DCT by 0.15 dB on average while DCT performs better on *Coastguard* by a margin of between 0.2 – 0.5 dB.

3.4 Wavelet Packets and Adaptive DCT Results

It is clear from previous section that the coding efficiency of a video compression system with scalar quantization scheme can be significantly improved by the use of linear transforms such as DCT and DWT. An improvement of 1.0 dB on average can be seen in most sequences for the transforms considered. However, the performance comparison between DCT and DWT on motion residual coding is less straightforward. The observations made can in general be categorized into two areas: transforms that perform better at low motion sequences and those that perform better at medium-high motion sequences. This leads to the obvious possible solution where one could choose a suitable transform by simply adjusting the tree-level in a dyadic DWT structure or the DCT size on a per picture basis based on some cost functions. While this is feasible in practice, it is unlikely to exploit all of the source redundancy present, or match the signal's non-stationary statistics. In this section, we will explore the use of a more flexible adaptive tree structure, using wavelet packets (WP) as a generalized wavelet decomposition for motion residual coding. See Chapter 2 for a more details discussion on the subject. The idea is to decompose the source using all possible wavelet packet bases of a given wavelet kernel, and then find the "best" wavelet packet basis. Coding results using DCT-based adaptive block (4×4 and 8×8) transform will also be given to facilitate the comparison of performance between the two classes of transform.

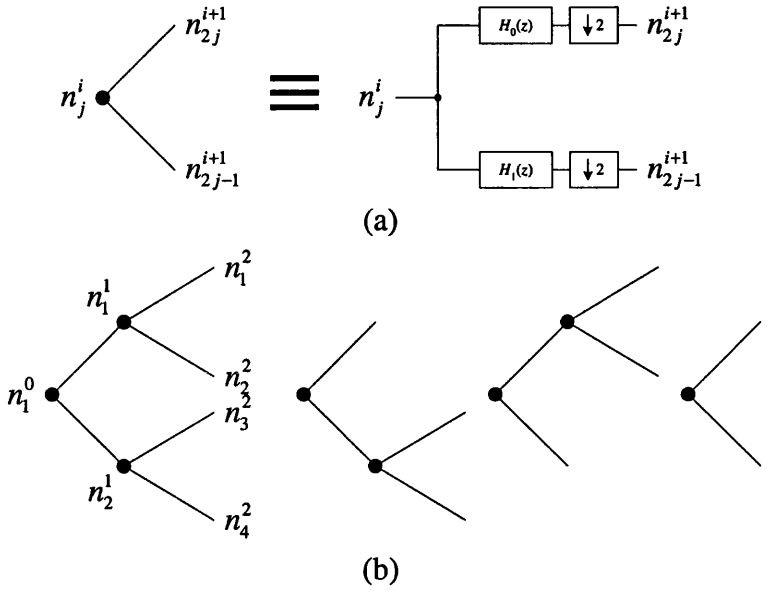


Figure 3.7 Tree notation for analysis filter bank (a) and all possible binary wavelet packet decompositions of depth 2.

To find the "best" wavelet packet basis for a signal block (of size N), a simple hierarchical extension method [94] to (3.19) is used. Let T refer to the finite set of nodes, $T = \{n_j^i\}$, of the full WP tree with a unique root node designated by n_1^0, where j (of possible 2^i choices) indexes the location of the node at the ith tree-depth or "scales" (for i = 1,2,...,log N). See Figure 3.7. Let S_j^i be a subtree of tree T rooted at the jth node of the ith tree-depth. If $i = 0$ and $j = 1$, then S_j^i is called a pruned subtree of T and we write $S_j^i \preceq T$. Let both rate R and distortion D be additive cost metrics over the WP tree, i.e. $R(S_j^i) = \sum R(\tilde{S}_j^i)$ and $D(S_j^i) = \sum D(\tilde{S}_j^i)$ where $\tilde{S}_j^i$ is the set of leaves of subtree S_j^i, and $q(n)$ denote the set of all admissible quantizers for node $n \in T$. Let $\{\hat{R}_j^i, \hat{D}_j^i, \hat{J}_j^i\}$ refer to the rate, distortion and Lagrangian cost associated with the optimum (for a given λ) subtree from n_j^i onward and split(n_j^i) be the decision of whether or not it is optimum to split the node into its children branches. The procedure for the complete algorithm is summarized as follow:

1. Compute full WP tree and gather both rate $R_q(n)$ and distortion $D_q(n)$ for all the nodes $n \in T \; \forall \; q \in q(n)$.
2. For a given Lagrange multiplier λ,
   i. Find $J_j^i(\lambda) \equiv J_n(\lambda) = \min_q D_q(n) + \lambda R_q(n) \;\; \forall \; n \in T$.
   ii. $i \leftarrow d$, where $d = \log N$ is the maximum signal block tree-depth.
   iii. Let q^* be the value of q that minimize $J_n(\lambda)$. Then for $n = n_j^d$, set

$$\left\{\hat{R}_j^d \leftarrow R_j^d = R_{q^*}(n); \hat{D}_j^d \leftarrow D_j^d = D_{q^*}(n); \hat{J}_j^d \leftarrow J_j^d\right\}$$

   iv. for $i = d\text{-}1, d\text{-}2, \ldots, 0$

       for $j = 1, 2, \ldots, 2^i$

           if $J_j^i(\lambda) < \hat{J}_{2j}^{i+1}(\lambda) + \hat{J}_{2j-1}^{i+1}(\lambda)$

$$\left\{\text{split}(n_j^i) \leftarrow \text{NO}; \hat{R}_j^i \leftarrow R_j^i; \hat{D}_j^i \leftarrow D_j^i; \hat{J}_j^i \leftarrow J_j^i\right\}$$

           else

$$\left\{\begin{matrix}\text{split}(n_j^i) \leftarrow \text{YES}; \hat{R}_j^i \leftarrow \hat{R}_{2j}^{i+1} + \hat{R}_{2j-1}^{i+1}; \\ \hat{D}_j^i \leftarrow \hat{D}_{2j}^{i+1} + \hat{D}_{2j-1}^{i+1}; \hat{J}_j^i \leftarrow \hat{J}_{2j}^{i+1} + \hat{J}_{2j-1}^{i+1}\end{matrix}\right\}$$

           end

       end

   end
3. Bisection search is used together with step 2 to find the optimum λ^* for a given target bit rate.

At the end of step 3, the element split(n_j^i) carves out the optimum pruned subtree $S^*(\lambda^*)$ and its associated optimum quantizers choice q^* for all the nodes $n \in S^*(\lambda^*)$. For adaptive DCT coding, the procedure is much simpler, as no tree pruning is needed. Given an arbitrary set of quantizers (e.g. identical to those used in the WP) and the Lagrange multiplier λ, the optimum Lagrangian cost associated with the best quantizer is calculated for both transforms (of sizes 4×4 and 8×8) that had been used on each signal block of size 16×16. The transform that exhibits a smaller Lagrangian cost is then chosen for the block.

Figure 3.8 shows the PSNR results from the adaptive DCT coder and the wavelet packet coder applied to the entire residual (with a tree-depth of 4) and

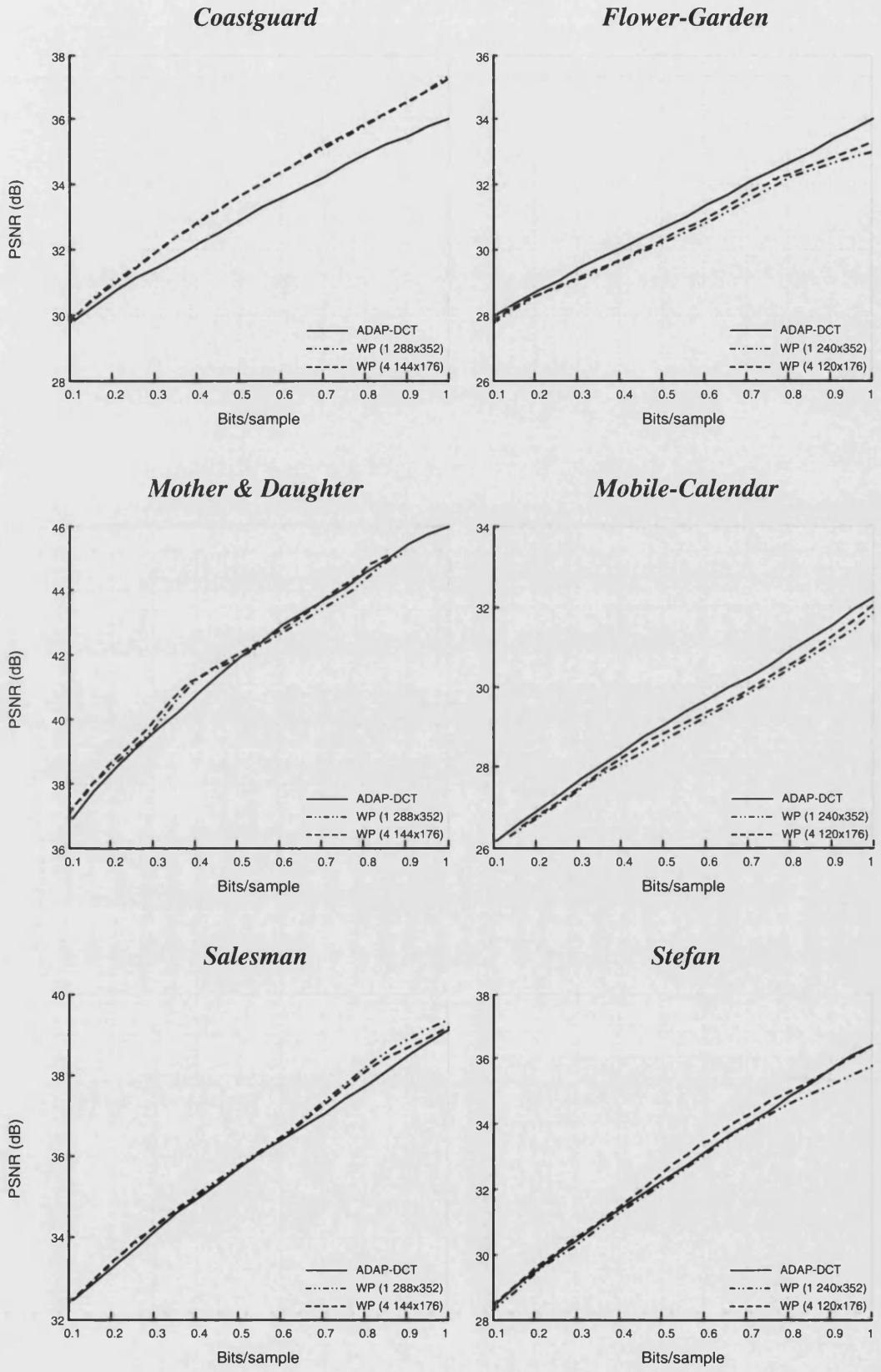


Figure 3.8 PSNR comparison of the DCT-based adaptive block coder and the wavelet packet coders using fixed time segmentation.

when the residual is divided into 4 nonoverlapping blocks (with a tree-depth of 3). Scalar quantizers with step sizes that halve with succeedingly deeper tree levels were chosen as members of the admissible set with the root's set being {32, 64, 96, 128}. From the figure, it can be seen that the PSNR performances from the three coders are fairly competitive with one another. The WP coder which operated on 4 quarter-sized blocks is in general 0.1 – 0.2 dB better than the one which operated on the entire residual. For *Mother & Daughter*, *Salesman* and *Stefan* sequences, the differences between the adaptive DCT coder and the WP coders are marginal. For *Flower-Garden* and *Mobile-Calendar*, the DCT coder outperforms the WP coders by 0.1 – 0.4 dB while the WP coder improved upon the DCT coder by as high as 1.4 dB on *Coastguard* at 1.0 bpp. It is interesting to note that the latter observation is opposite to those obtained in the previous experiment (multiple quantizers scheme) where fixed size DCT achieved higher PSNR than DWT on *Coastguard* while DWT did better on *Mobile-Calendar*. A manual check between the two sets of results reveals that the adaptive DCT coder always achieves equal or better PSNR performance compared to its best fixed size counterpart while the WP coder achieves very significant PSNR gain on *Coastguard*, moderate gain on *Mother & Daughter*, *Salesman*, *Stefan* and *Flower-Garden*, but 0.4 dB worst on *Mobile-Calendar* compared to the fixed structure DWT.

3.5 Summary

In this chapter, several characteristics of the motion compensated residual have been exemplified and the coding efficiency of the DCT and DWT in residual coding examined in both a theoretical and practical setting. Experiments conducted include statistical modeling using Generalized Laplacian Distribution to obtain the rate-distortion performance bounds and two scalar quantization schemes to gather the operational performance bounds of both transforms. Key observations from the experiments can be summarized as follow:

1. Employment of linear transforms in a video compression system using simple scalar quantization scheme can provide 0.5 – 1.0 dB improvement in

peak signal-to-noise ratio. The added performance gain varies with the type of transform used, the coding rate and residual content.

2. DWT (using 9/7, 5/3 or the haar filters) with 1 – 2 levels of decomposition is sufficient for motion residual coding. Additional levels of decomposition provide little, if any, performance gain.
3. DWT (using 9/7, 5/3 or the haar filters) does not provide any significant gain over the DCT in motion residual coding.

Finally, a simple best basis selection exercise has demonstrated that adaptive transforms can achieve more consistent results in term of peak signal-to-noise ratio in motion residual coding as compared to their fixed structure counterparts. The performance differences between both DCT- and DWT-based adaptive coders are only marginal.

Chapter 4

Embedded DCT Coding Using Significance Block Map

4.1 Introduction

Motion estimation and compensation is widely employed in many video compression schemes to exploit the high degree of correlation between adjacent pictures. In its earliest implementations, motion estimation was carried out in the spatial domain with fixed size blocks. Later developments have seen the technique being applied in the transform domain and in segmentation-based schemes, where motion estimation and compensation is performed on the basis of regions.

Although encouraging results have been reported with these new implementations, all techniques based on motion-compensation suffer from the impossibility of reconstructing new objects. To remedy this drawback, compensation images that are usually called displaced frame differences or simply residuals have to be encoded and transmitted as well. A common practice in many video compression systems is to compress these error signals using simple variants of still image compression algorithms. For example, both H.26x and MPEG use a flat quantization table with the discrete cosine transform for inter-frame coding whereas visually tuned quantizers are used for the intra-frame coding. Such an arrangement however is unlikely to produce optimum results due to the different image characteristics.

In this chapter, we describe an embedded coding technique for motion residuals that are obtained via time-domain fixed (or variable) block size motion compensation schemes. As with all embedded coding algorithms, the embedded property allows precise control of coding rates since the bit-stream can be truncated at any point according to either rate or distortion constraints. In particular, we show that better coding performance in term of PSNR, as well as improved visual quality, can be achieved with the discrete cosine transform by using a better strategy for coefficient organization.

4.2 Overview of algorithm

Figure 4.1 shows a simple block diagram of the proposed encoder. Frames of a video are assumed to be processed by a typical block-based motion estimation scheme (e.g. MPEG) and the resulting motion compensated residuals encoded by the new encoder. The structure of the residual coder follows the three-model para-

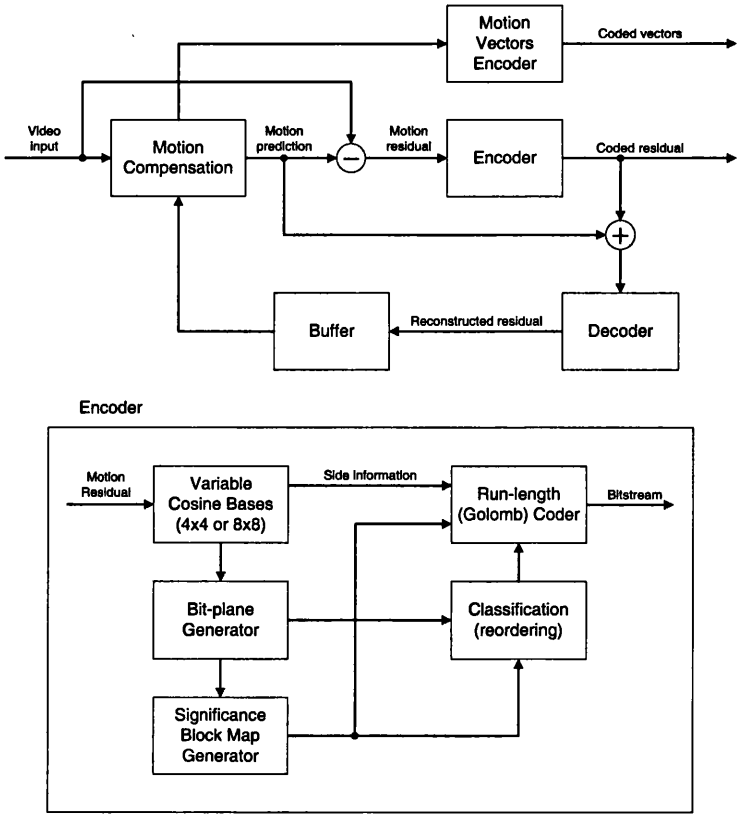


Figure 4.1 Block diagram of the embedded encoder.

digm depicted in Figure 1.3: a linear transform based on cosine bases, a set of uniform quantizers and a run-length entropy coder that is based on Golomb codes. DCT of sizes 4×4 and 8×8 are used to transform the motion residuals into the frequency domain. The transform coefficients are then encoded in bit-plane order, or equivalently, quantized by a series of successively refined uniform scalar quantizers with step sizes $\Delta_n = 2^n \Delta$, where Δ is the finest quantization step size. The encoding of each bit-plane, or quantization indices, is driven by a significance block map that is created to take advantage of the clustering of large (or small) variance coefficients inherent in motion-compensated residuals. A simple classification scheme based on the significance block map is then applied to the identified significant coefficients to improve coding efficiency. Both the significance block map and the ordered binary coefficients bit-stream are entropy coded by a simple run-length coder and multiplexed to form the embedded bit-stream. The significance map is then updated at the next bit-plane and the same process repeated.

Experimental results show that the proposed method outperforms MPEG-2 and H.263 in coding of inter-frames by 0.3 – 2.5 dB and 0.05 – 1.26 dB respectively. In most cases it also outperforms the wavelet-based JPEG-2000 coder and arithmetic entropy coding version of SPIHT, hereafter referred to SPIHT-AC, and has lower complexity. Blocking artifacts are also less visible in some sequences coded at low bit rates.

4.3 Significance Block Map

The significance block map is a bi-level structure that provides a means of coding zeros jointly. It has a fixed size and fixed support, i.e. the number of coefficients associated with each element of the map is constant. As will be seen in the later section, the significance block map can also be used as a simple classification mechanism whereby coefficients that are more probable to be found significant in the next encoding pass are coded first, with no side information needed.

Before we proceed to the definition of the significance block map, it is necessary to introduce some notations to aid in our analysis. Let the source x be divided into 16 × 16 spatial blocks B_i, $1 \le i \le N$, where B_i is scanned from left to right, top to bottom fashion. Each block B_i is transformed to the frequency domain using DCT of sizes 4 × 4 or 8 × 8. The actual DCT size to use will be discussed in a later section. Let $c_{i,j} \in B_i$ refer to the set of DCT coefficients in block B_i, where j indexes the position of the coefficient in some scanning order. Let $b_{i,j}^k b_{i,j}^{k-1} ... b_{i,j}^0$ denote $|c_{i,j}|$ in binary, where k is the smallest integer satisfying $2^{k+1} > |c_{i,j}|$ for all i and j. Let $P^n = \bigcup_{i,j} b_{i,j}^n$ of the same dimension as x denotes the nth bit-plane of the transform coefficients. For a given P^n, $n = k,k-1,\ldots,0$, form the nth *significance block map* $M^n = \{m_i^n\}$ according to

$$m_i^n = \begin{cases} 0, & b_{i,j}^n = 0, \ \forall j \\ 1, & otherwise \end{cases}, \quad i = 1,2,...,N, \tag{4.1}$$

and the corresponding *significance sequence*

$$S^n = \{b_{i,j}^n \mid m_i^n = 1; \ \forall i,j\}. \tag{4.2}$$

Here, $S^n \subset P^n$ is a collection of blocks of binary map $\in \{0,1\}$ where at least one significant coefficient exists in the spatial location indicated by each map. A graphical illustration for most definitions can be found in Figure 4.2. Both M^n and

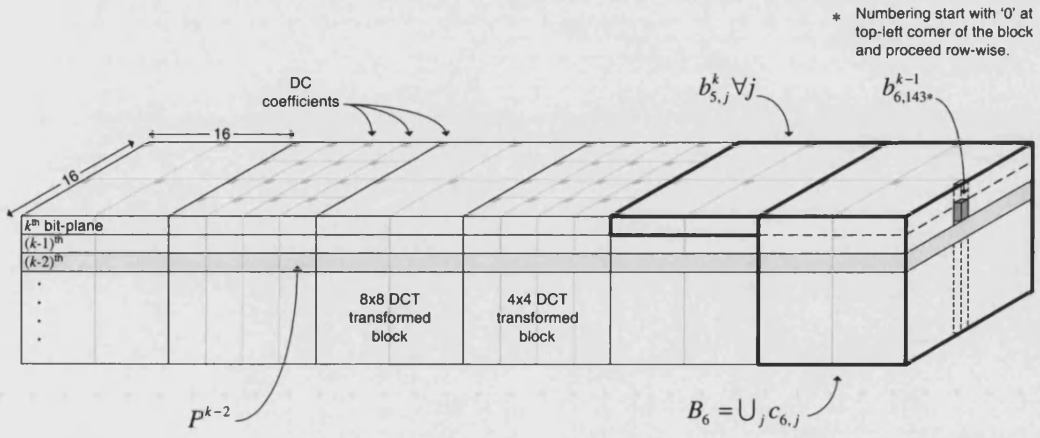


Figure 4.2 Definitions of b_i and c_j^i. Six pixel blocks (e.g. $N = 6$) are shown and all but b_2 and b_4 are transformed by 8×8 DCT.

S^n are then rearranged according to Section 4.5 of this chapter and run-length coded. The significance block map is then updated and the same process repeated for $n = k$-1. Note that for $n < k$, the significance block map M^n only needs to be updated for $\{m_i^n \mid m_i^{n+l} = 0\}$, where $l = 1, 2, ..., k-n$. In other words, once m_i^n if found significant, $m_i^{n-1} = m_i^{n-2} = ... = m_i^0 = 1$ and are not entropy coded.

In order to improve the compression ratio of M^n, a simple form of classification by context is used. The probability of the occurrence of zeros in M^n is partitioned into four non-overlapping intervals and the interval in which M^n belongs is communicated to the decoder as side information. The four intervals are defined as

1. $\Pr\{m_i^n = 0\} = 1.0$
2. $1.0 > \Pr\{m_i^n = 0\} \geq 0.5$
3. $0.5 > \Pr\{m_i^n = 0\} \geq 0.1$
4. $\Pr\{m_i^n = 0\} < 0.1.$

Since there are no changes in M^n in interval 1, i.e. $M^n \equiv M^{n+1}$, it is not coded. In interval 4, we have a complementary condition to interval 1, where the advantage of sending the significance map eventually diminishes at high rates, i.e. $\Pr\{m_i^n = 0\} \rightarrow 0$. It was found empirically that when the probability of a zero in M^n falls below 0.1, there is no gain in coding M^n and that the budget is better spent on coding the significance sequence S^n instead. Consequently, M^n is also skipped in this mode. Intervals 2 and 3 act as simple binary events in which the *more probable symbol* (MPS) used in the adaptive run-length coder (see Section 4.6) is "toggled". More specifically, when the probability of a zero in M^n falls below 0.5, the MPS is set to '1' or else '0' is used instead. Note that when the adaptive run-length coder is used with S^n, the MPS will always be set to '0'. This is because the probability of a zero in these sequences always satisfies $\Pr\{m_i^n = 0\} > 0.5$ at bit rates of practical relevance.

4.4 Fast cosine bases selection

We saw in the previous chapter that adaptive solutions that take into account the source statistics provide more consistent results in term of PSNR performance, as compared to transforms of fixed sizes (from DCT) or implementations (e.g. tree level in a dyadic DWT decomposition). Although the optimization structure in which those results are obtained is also applicable in our current context, it suffers from two drawbacks:

1. The method does not produce an embedded bit-stream in its original implementation.
2. High computation cost.

It is worth nothing that the latter condition by itself does not constitute the main obstacle that prevents the use of the Lagrange multiplier method, or any optimization techniques for that matter, in a video compression system; it is the collective cost from all core components (e.g. motion compensation using multiple frames and higher resolution pel accuracy) that formed the compression system that raises the complexity issue. Therefore, in our current implementation, we seek a fast and low cost solution to incorporate different cosine bases. One such algorithm is to use simple first order statistics such as entropy and variance to determine the DCT size to use.

To aid in the assessment of the merit of the various statistics, a restricted set (i.e. step sizes are power of 2) of UQDZ quantizers and the Lagrange multiplier method were used to find the best mode for each 16×16 block at 10 different bit rates that are equally spaced between 0.1 – 1.0 bpp. Note that the restriction on the quantizers step size to be power of 2 is necessary to simulate bit-plane coding of each block. These decisions then serve as a reference and the number of "correct" selections made by each test statistic recorded. Figure 4.3 shows the percentage of "correct" selections made by the six test statistics. The calculation for *Entropy* and *Variance* is as the name suggested; *AbsSum* designates the sum of the absolute values of the transform coefficients and *AST4–12* are similar to *AbsSum* except that the coefficients are first hard-thresholded by the integers listed with the names. The decision on which transform to use is then based on the value calculated for each

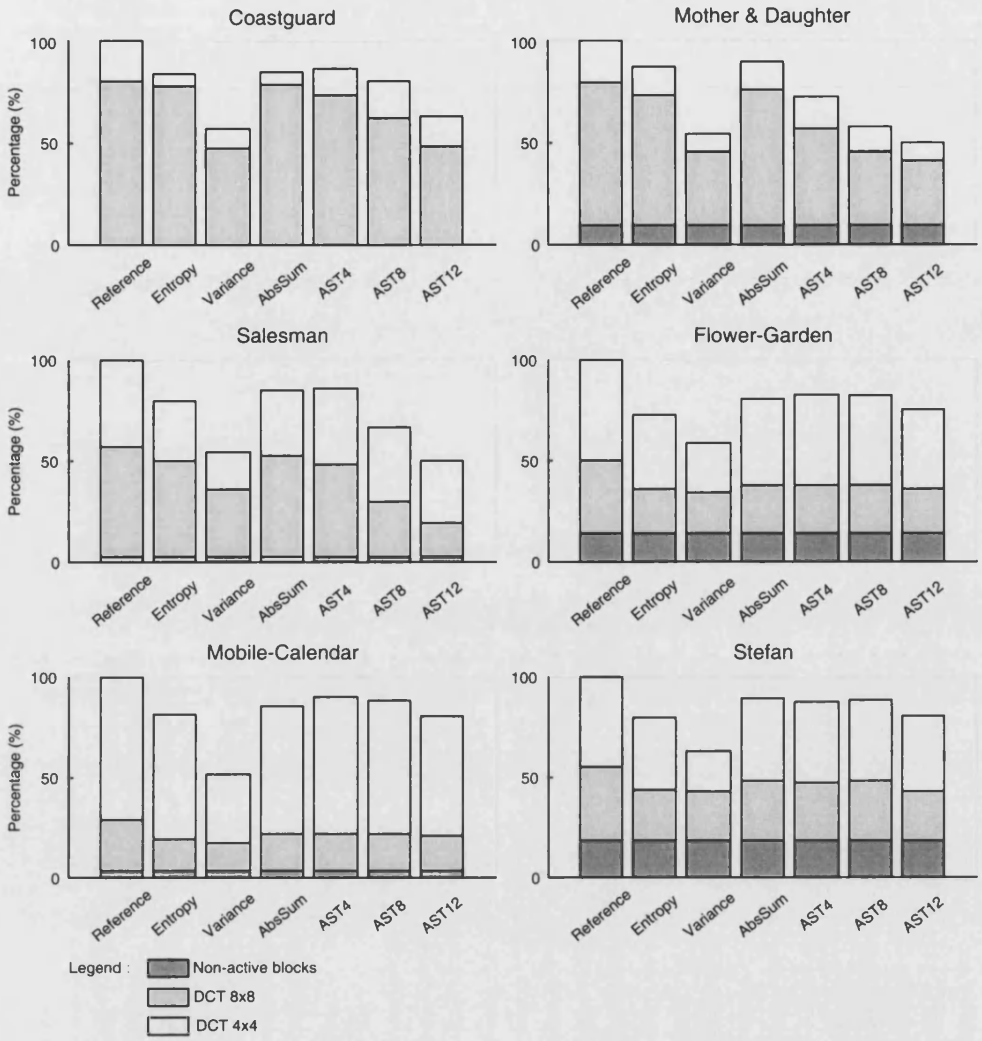


Figure 4.3 Percentage of "correct" selections made by six test statistics, measured with respect to the Lagrange multiplier method.

transform: for *Entropy*, *AbsSum* and *AST4–12*, the transform that exhibits a smaller value is used, while a transform with a larger value is used for *Variance*. It can be seen from the figure that *AST4* and *AbsSum* offer the most "correct" selections in most test cases, followed by *AST8* and *Entropy*. The objective performance (e.g. peak signal-to-noise ratio) between the four statistics were empirically found to be marginal for most common test sequences coded at 128 – 2560 kbits/s, with *AST4* and *AbsSum* outperforming *AST8* and *Entropy* by 0.05 dB. *AST4* however was found to be less sensitive to the motion compensation technique employed (e.g. standard BMC or overlapped BMC) in the test system as compared to *AbsSum*, and is therefore used in our implementation to choose the DCT size. Formally stated,

the transform coefficients from the two alternatives are pre-processed by the operator

$$\hat{c}_{i,j} = \begin{cases} 0, & -4 \le c_{i,j} \le 4 \\ c_{i,j}, & otherwise, \end{cases} \tag{4.3}$$

and compared using the criteria

$$\alpha_i = \sum_j |\hat{c}_{i,j}|. \tag{4.4}$$

The transform that exhibits a smaller α is used for the ith block.

4.5 Coefficients Reordering

To improve coding efficiency, a reordering scheme similar to [83] is applied to both M^n and S^n. Let R^n refer to the set of coefficients found significant in the previous k-n iterations, i.e. $\{c_{i,j} \mid b_{i,j}^{n+l} = 1\}$, where $l = 1, 2, ..., k-n$. Let I_i refer to the set of DCT size selection indices for the spatial blocks B_i, i = 1,2,…,N. For each bit-plane $n \le k$, we rearrange $\{m_i^n\}$, $\{p_{i,j}^n\}$ and R^n to form five subsequences s_q, q = 1,2,…,5, as follows:

1. *Past significant*: $s_1 = \{b_{i,j}^n \mid m_i^{n+1} = 1;\ \forall \{i,j\} \notin R^n\}$
2. *Significance Map*: $s_2 = \{m_i^n \mid m_i^{n+1} \neq 1\}$
3. *Selection indices*: $s_3 = \{I_i \mid m_i^n = 1\}$
4. *Current significant*: $s_4 = \{b_{i,j}^n \mid m_i^n = 1\} \cap \{b_{i,j}^n \mid m_i^{n+1} \neq 1\}$
5. *Refinement*: $s_5 = \{b_{i,j}^n \mid \{i,j\} \in R^n\}$.

Note that S^n has been partitioned into three disjoint sets, i.e. $S^n = s_1 \cup s_4 \cup s_5$, based on R^n and the previous and current significance map M^{n+1} and M^n. The advantages of this partitioning are twofold: 1) it delays the coding of s_2 and 2) creates two subsets with the characteristic that $\Pr\{s_1 = 1\} \ge \Pr\{s_4 = 1\}$ is observed for all test sequences examined. The former condition is beneficial because there is no reduction in MSE despite the fact that some bit budgets are used while the latter consolidates zeros and hence improved the coding efficiency of the entropy coder. Another im-

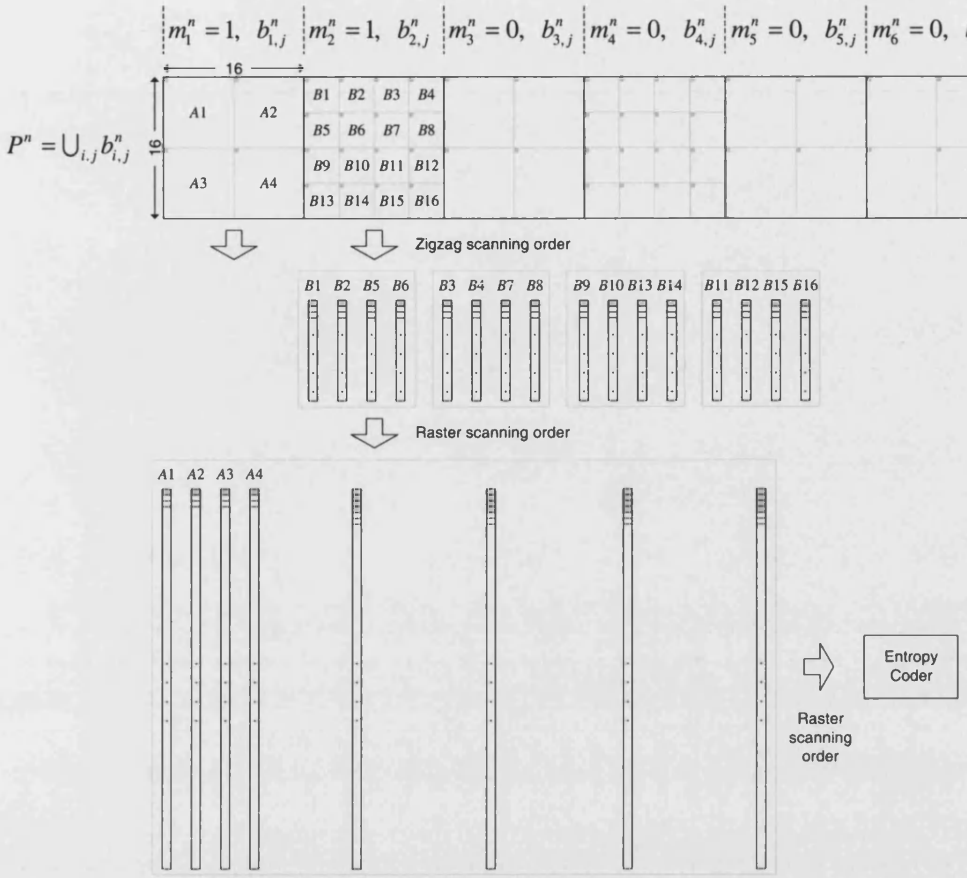


Figure 4.4 Scanning order for subsequences s_1 and s_4.

portant feature in our ordering is that we only send the DCT size selection indices for $\{m_i^n\}$ that is switched on during the current pass. In this way, we reduce the side information that is needed to specify the DCT size selection and waste no bits on information that the decoder does not need. We note that s_1 can be further separated into two disjoint sets based on the neighborhood of the significant coefficients in the set R^n [83]. The improvement, however, is only marginal.

Each subsequence s_q is separately entropy coded in the order they are numbered, except for s_5, which is appended to the bit-stream uncoded. Simple raster scanning order is used in both s_2 and s_3. The order in which s_1 and s_4 are scanned is illustrated in Figure 4.4, using the example given in Figure 4.2. Each DCT block in the significant $\{b_{i,j}^n\}$ is first zigzag-scanned to produce a column-wise vector. No further action is required for vectors of length 64, i.e. those from 8 × 8 DCT. For

vectors of length 16, they are grouped into small groups of 4 such that vectors in the same group come from the same quadrant in each 16 × 16 block, and raster-scanned to form another column-wise vector but of length 64. These new vectors are then "merged" with those from the 8 × 8 DCT and raster-scanned to form the final single vector that is to be entropy coded. When a symbol of '1' is encountered in the entropy coder, the sign bit of the coefficient at the same location is appended uncoded to the bit-stream following the encoding of the symbol. After a target rate has been reached, all significant coefficients are half-adjusted such that

$$b_{i,j}^{\hat{k}-1} = 1, \tag{4.4}$$

where $\hat{k}$ is the last bit-plane to be coded.

4.6 Golomb Code

Golomb code [31] is a parameterized family of countably infinite codes that is designed to encode nonnegative integers with a geometric probability assignment

$$P(i) = (1-\theta)\theta^i, \quad i \geq 0, \tag{4.6}$$

for some arbitrary θ, $0 < \theta < 1$. This particular distribution arises in run-length coding, where if one has an independent letter binary source, with θ being the probability of a zero, then $P(i)$ is the probability of a run of i zeros. The key to the parameterized family of codes is that ranges of probability value $\theta \geq \frac{1}{2}$ are converted to a single parameter characterized by a positive integer $m \geq 1$, such that the approximation

$$\theta^m \approx \tfrac{1}{2} \tag{4.7}$$

is satisfied.

Corresponding to each parameter m and run-length r is a two-components codeword:

1. unary or base-1 codeword for quotient $Q = \lfloor r/m \rfloor$
2. codeword for remainder $R = r \bmod m$.

The unary codeword is of the form '0' or more '1s' followed by a delimiting '0' that signals the end of the number for quotient Q. The remainder R is encoded by a code that is dependent upon the parameter m. When m is a power of 2, e.g. 2^k, the code for R is the k-bit binary number representing the remainder R, and so there are 2^k code-words. When m is such that $2^{k_{max}} < m < 2^{k_{max}+1}$, then there are $g = 2^{k_{max}+1} - m$ and $h = 2m - 2^{k_{max}+1}$ code-words of k_{max} and k_{max}+1 bits respectively. The g code-words of length k_{max} is given by the first g code-words of $m = 2^{k_{max}}$ while the last $h/2$ code-words of $m = 2^{k_{max}}$ are "extended" to form the h code-words. For example, if m = 6, then k_{max} = 2 and the last (12-8)/2 code-words for m = 4, which are 10 and 11, are extended respectively to 100 and 101 from 10, and 110 and 111 from 11. The vector set {00,01,100,101,110,111} thus defines the codes for R of m = 6. Table 4.1 lists the run-length dictionary for $4 \leq m \leq 8$, where each codeword is separated into the unary and remainder parts. To reconstruct the run-length r, the quotient Q, which is the number of '1s' preceding the first '0' from the left, is multiply by the parameter m and add to the remainder R, e.g. $r = m \times Q + R$. For $m = 2^k$, the code for R is simply the next k-bit after the delimiting '0'. For $m \neq 2^k$, a k-bit look-up table is often used for decoding.

Table 4.1 Golomb codes for $4 \leq m \leq 8$.

m	4		5		6		7		8	
r	Q	R	Q	R	Q	R	Q	R	Q	R
0	0	00	0	00	0	00	0	00	0	000
1	0	01	0	01	0	01	0	010	0	001
2	0	10	0	10	0	100	0	011	0	010
3	0	11	0	111	0	101	0	100	0	011
4	10	00	0	110	0	110	0	101	0	100
5	10	01	10	00	0	111	0	110	0	101
6	10	10	10	01	10	00	0	111	0	110
7	10	11	10	10	10	01	10	00	0	111
8	110	00	10	111	10	100	10	010	10	000
9	110	01	10	110	10	101	10	011	10	001
10	110	10	110	00	10	110	10	100	10	010
11	110	11	110	01	10	111	10	101	10	011
12	1110	00	110	10	110	00	10	110	10	100
13	1110	01	110	111	110	01	10	111	10	101
14	1110	10	110	110	110	100	110	00	10	110
15	1110	11	1110	00	110	101	110	010	10	111

Gallager and Voorhis have shown in [33] that the Golomb code for a particular value of m is optimum not only when (4.7) is satisfied, but more generally for θ satisfying

$$\theta^m + \theta^{m+1} \leq 1 < \theta^m + \theta^{m-1}. \tag{4.8}$$

This inequality condition sharpens (4.7) and identifies the precise breakpoint probability values θ that separate successive m and m+1. In other words, it enables one to select an optimum parameter m if one has *a prior* knowledge of the probability of the source $x \in \{1,0\}$. Let '0' be the *more probable symbol* (MPS) and therefore '1' the *less probable symbol* (LPS). Table 4.2 shows the probabilities range for p(LPS) that were calculated numerically for a given value of m by using test values of θ until equality was almost reached in the left half of (4.8). As implied by the notation, the ranges are closed on the high-valued probability boundary, and open on the low-valued boundary and are therefore non-overlapping. As an example, consider encoding a vector of run lengths {7,18,11,21,4,15}. There are 6 LPS and therefore p(LPS) = 0.0789 and p(MPS) = 1-p(LPS). Thus from Table 4.2, m = 8 should be used.

The selection of m based on (4.8) requires m to be sent as side information on a predetermined interval. From a coding point of view, this takes up bits and it is not clear how the interval should be selected to maximize the coding efficiency.

Table 4.2 Ranges of p(LPS) for $1 \leq m \leq 10$.

m	Probability range for the *stop* event
1	(0.61803398874989 – 0.38196601125011)
2	[0.38196601125011 – 0.24512233375331)
3	[0.24512233375331 – 0.18082748660384)
4	[0.18082748660384 – 0.14332511614550)
5	[0.14332511614550 – 0.11872853836643)
6	[0.11872853836643 – 0.10134628737130)
7	[0.10134628737130 – 0.08840764651795)
8	[0.08840764651795 – 0.07840068036602)
9	[0.07840068036602 – 0.07042987176798)
10	[0.07042987176798 – 0.06393088892224)

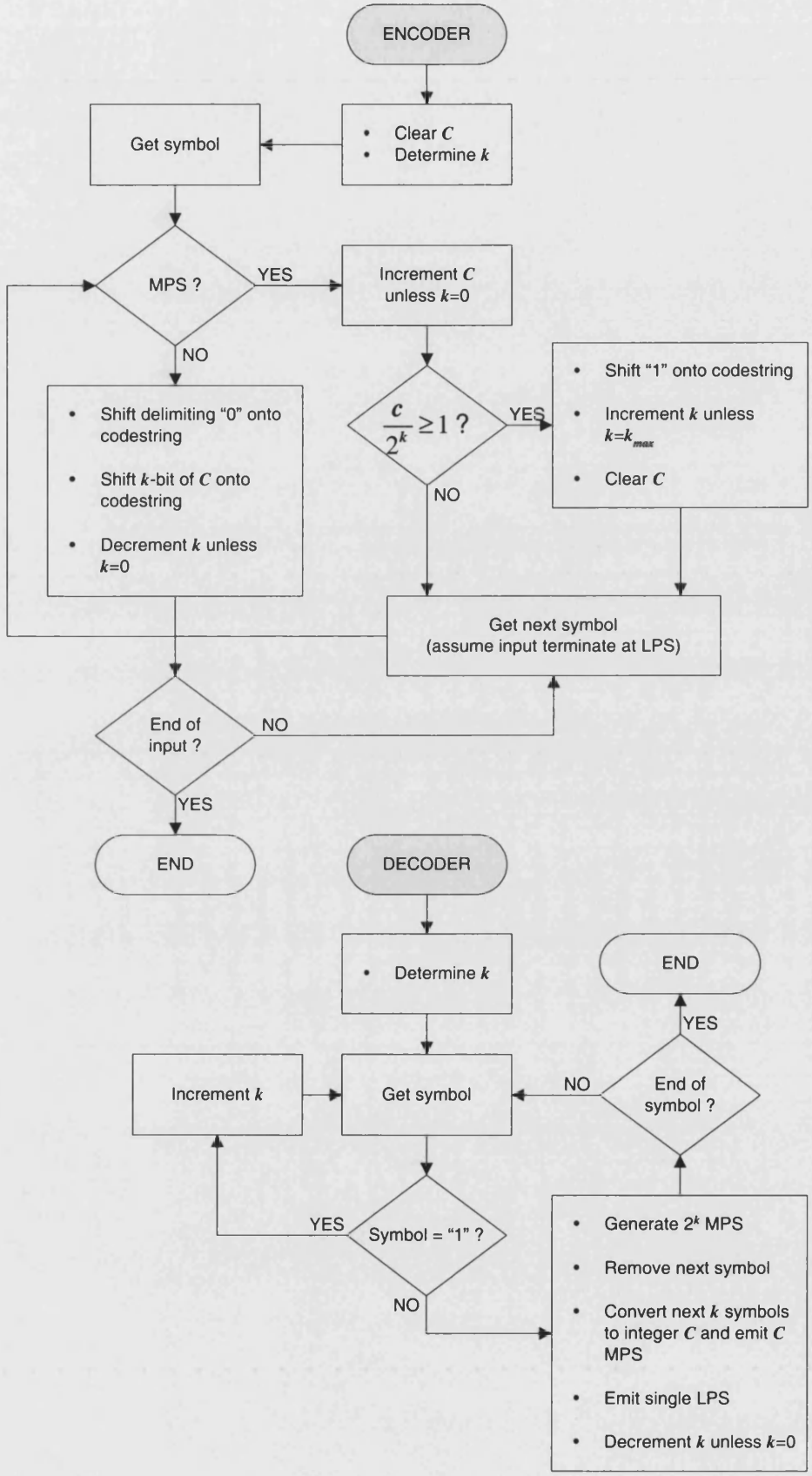


Figure 4.5 Details of an adaptive Golomb encoder and decoder.

Another alternative and better method is to adapt m as bits are encoded. Several adaptation strategies [57, 83] have been developed and the algorithm in [57] is used to encode the subsequences s_1–s_4 in this work. Both the encoder and decoder operations are shown in Figure 4.5. Values of m that are powers of 2 are used for simple implementation. The algorithm exploits the insight $p(\text{MPS})^m = 1/2$ as a balance point for adapting the Golomb parameter m either up or down. For geometrically distributed run lengths, the probability that the run length is m or greater is $p(\text{MPS})^m$. For Golomb codes for binary sequences for which the parameter m is the correct value, run length m is such that half the runs are expected to be of median length m or more. Therefore, the algorithm takes m MPSs before the code adds a stone-age 1, so the first bit of the Golomb code indicates whether the length is 0 to m-1, or is m or greater. If at a particular parameter m, the length of the run is shorter than expected then the parameter m is reduced. If the run is longer than expected then the parameter m is increased. Thus the run length serves as a test in the algorithm, if the algorithm determines the run exceeds the current value of m then a branch toward a more highly skewed estimate is made. If the run ends then the mod m counter has a count less than m and the algorithm determines a less skewed value for m. Also, adaptation only takes place during code length increases, when a stone-age 1 is added, or when the run is ended by the occurrence of a LPS.

4.7 MPEG-2, H.263 and Wavelet-based Coders Comparison Results

In our first experiment, we compared the proposed residual encoder to standard DCT coding by a quantization table. These were produced with the publicly available MPEG2 version 1.2 software encoder [75]. To allow close comparison with MPEG, we have created a new video encoder by simply replacing the B- and P-pictures coding routines in MPEG with our new embedded encoder so that I-pictures coding, motion estimation/compensation and motion type/vectors encoding were performed in the same fashion in both systems.

The performance comparison was carried out on the first 50 frames in six common test sequences: *Akiyo*, *Salesman*, *Flower-Garden*, *Football*, *Mobile-Calendar* and *Stefan* in either CIF (352 × 288) or SIF (352 × 240) resolution. A GOP pattern of I–B–B–P–...– I–B–B–P was used in *Flower-Garden*, *Football*, *Mobile-Calendar* and *Stefan* while the so-called low delay mode I–P–P–P–...–I–P–P has been used in *Akiyo* and *Salesman* sequences. Coding was performed on the luminance component only. To synchronize the bit rate between the two systems, the MPEG encoder is first run using a target bit rate and frame rate. A record is kept of the number of bits (minus the number of bits used to specify the changing quantization step size) used to represent each frame. The modified encoder then uses this record to encode the same set of original frames with the same number of bits for each frame. Note that if a macroblock in B- and P-pictures in the modified system is to be intra-coded, it is compressed using the same coder without any modification.

Figure 4.6 and 4.7 show the inter-frame PSNR results from MPEG and the proposed coder using fixed 8 × 8 and variable DCT. It can be seen from the figures that both the proposed embedded coders improved upon MPEG by a significant margin on all test sequences. The average improvement attained for both coders is between 0.3 – 2.5 dB depending on the sequences and bit rates. Notice that the sudden dip in PSNR around frame 25 for *Akiyo* and *Salesman* is due to the start of a new GOP. For the remaining sequences, GOP starts at frame 1, 12, 24, 36, 48 and frame 1, 15, 30, 45 for CIF and SIF respectively. The differences between the two proposed coders are only marginal for *Flower-Garden*, *Football* and *Stefan*. For the remaining sequences however, the coder using variable DCT outperforms the fixed 8 × 8 DCT coder by 0.1 – 0.7 dB on average, with the largest improvement coming from the *Mobile-Calendar* sequence. The subjective quality resulting from the proposed coder is also slightly better for some sequences coded at low to medium bit rates; in particular, blocking artifacts are reduced and edges of moving objects are rendered much better (see Figure 4.10).

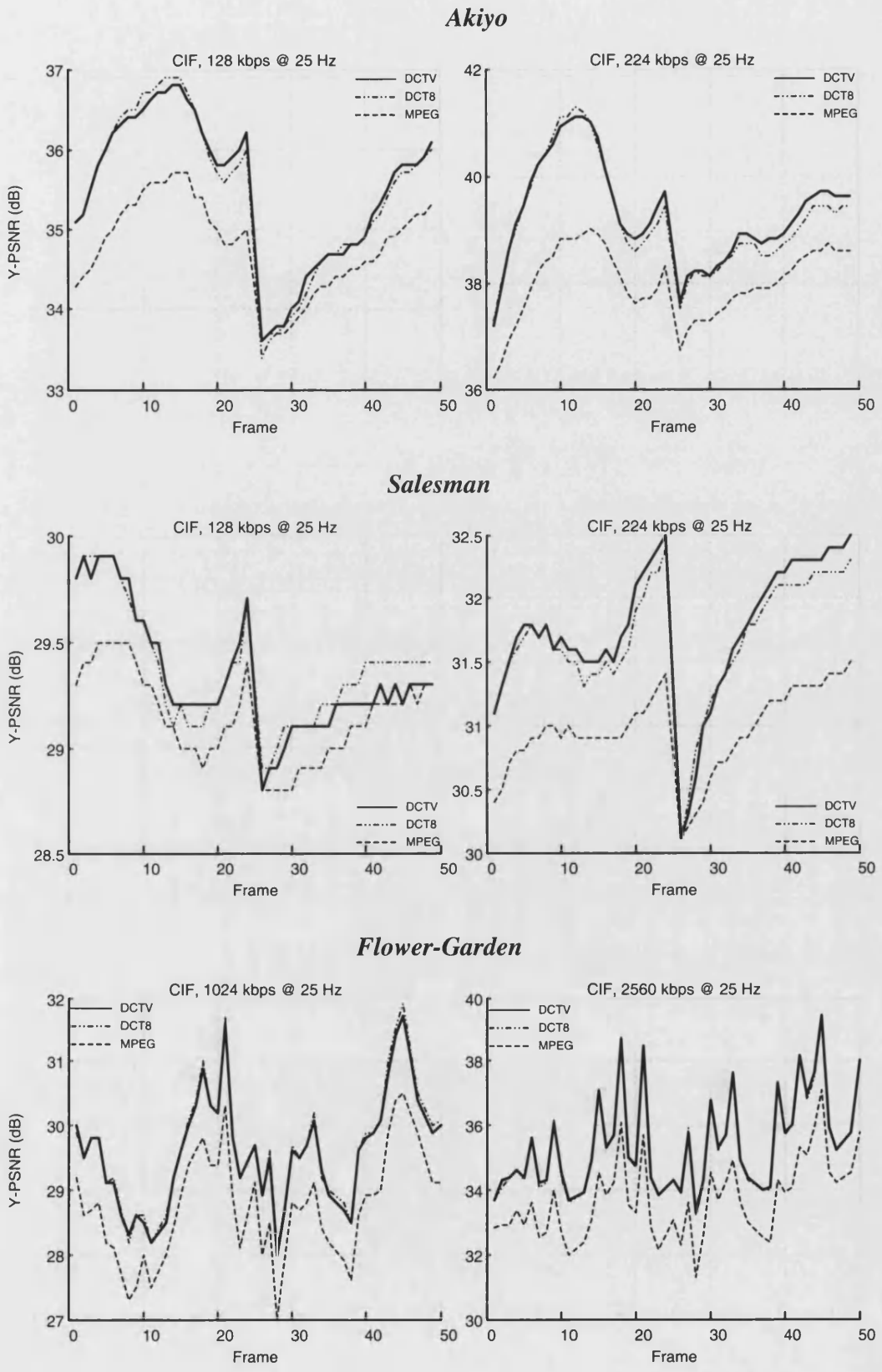


Figure 4.6 MPEG PSNR comparison for *Akiyo* and *Salesman* at 124 and 224 kbits/s and *Flower-Garden* at 1024 and 2560 kbits/s.

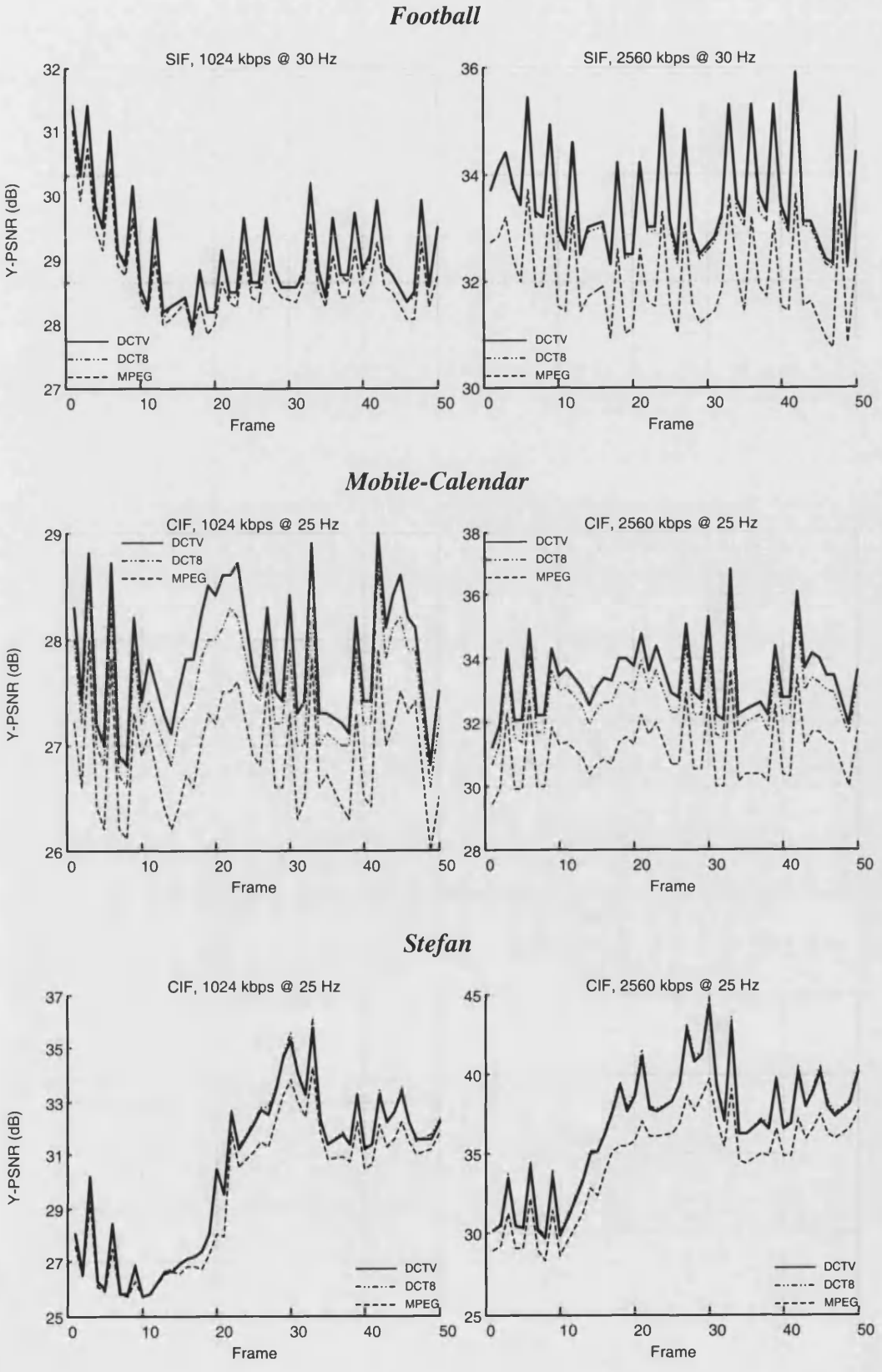


Figure 4.7 MPEG PSNR comparison for *Football*, *Mobile-Calendar* and *Stefan* at 1024 and 2560 kbits/s.

In our next experiment, we compared the wavelet-based SPIHT-AC coder [104], which has also been shown in [93] to work better than the DCT on motion residuals, and the JPEG-2000 coder [48, 71] to our embedded coder (variable DCT coder). The focus here is to acquire a better insight into the coding of motion residuals with cosine and wavelet transforms. For this particular experiment, we have used the H.263-based software encoder in [78] as a reference instead. The reasons for using the H.263 model are twofold: 1) it provides higher resolution accuracy and smaller block size in the estimation process and hence produced smoother residuals that require fewer bits to code, and 2) allows the use of a deblocking filter on the compensated picture and an advanced prediction technique where the mean of intra predicted macroblocks is removed prior to filtering. The latter is crucial to a global transform such as the DWT, because the sudden increase in intensity at block boundaries will create huge amounts of high frequency components when filtered and therefore decrease the coding efficiency. The experiment was also performed on the first 2 seconds of six sequences but with *Akiyo*, *Salesman* and *Football* replaced by *Coastguard*, *Mother & Daughter* and *Foreman* because they can be encoded at very low bit rates (using QCIF resolution) with acceptable visual quality. The actual frame size, frame rate and bit rates used for each sequence are indicated at the top of each plot in the figures. The first frame in all sequences was coded using the standard H.263 DCT intra-mode coding technique. All remaining frames in the sequence are motion-predicted and coded using the respective coders, i.e. the two seconds are coded as one long I–P–P–P group of pictures (GOP). Coding was performed on the luminance component only and the bit rates between different coders are synchronized using the same technique as specified in the previous experiment.

Figure 4.8 and 4.9 shows the inter-frame PSNR coding results from the three coders. To maintain figures readability, only JPEG-2000 with two levels of decomposition (JP2) is shown. Note that JP2 achieved the best average PSNR performance in all test sequences when compared to both JP1 and JP3. Also, single layer syntax is used in the JPEG-2000 coder for maximum coding efficiency. From the figures, it can be seen that the proposed DCT-based coder outperforms both wavelet-based methods typically by the order of 0.1 – 1.0 dB in peak signal-to-

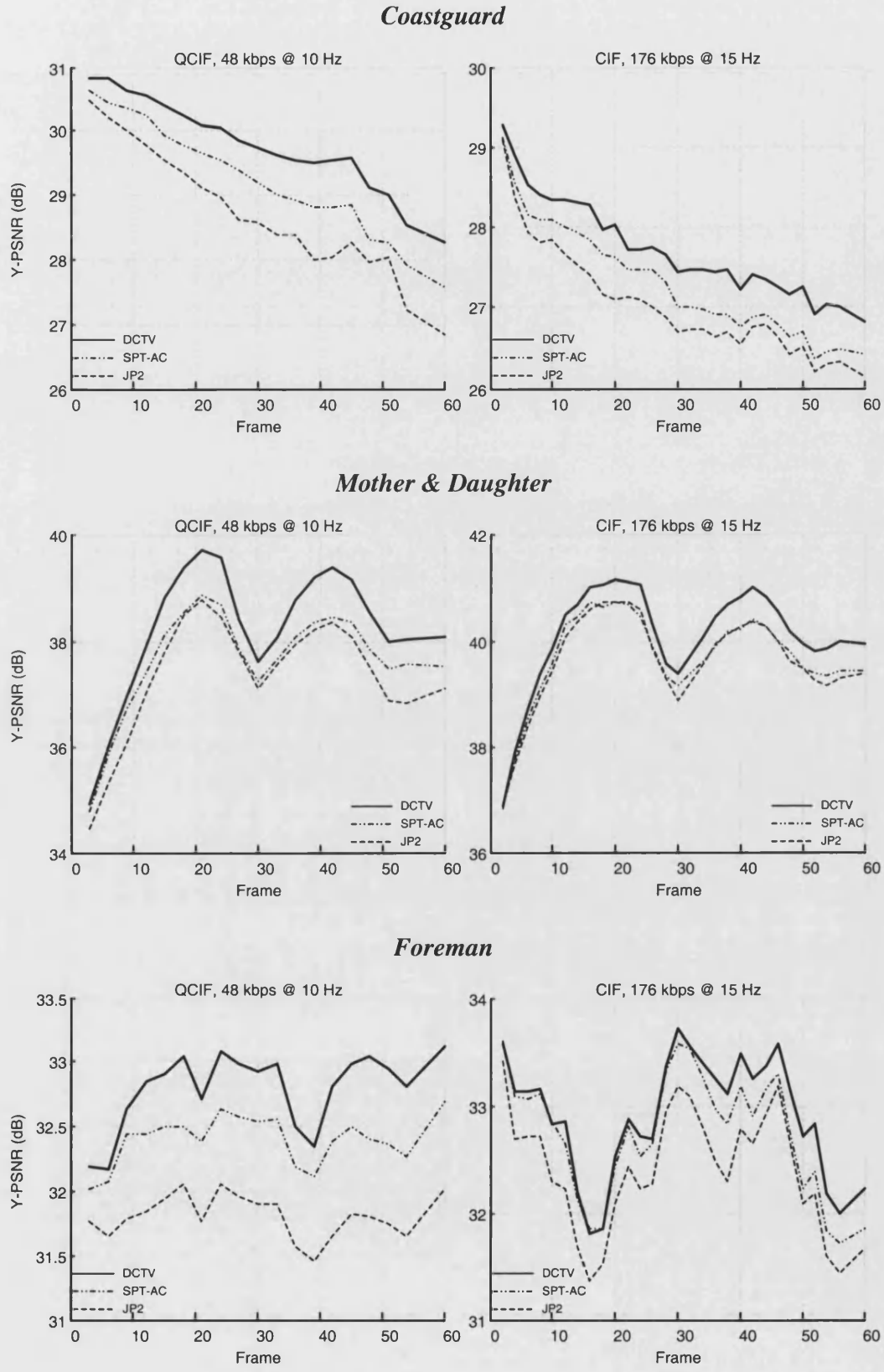


Figure 4.8 Wavelet-based PSNR comparison for *Coastguard*, *Mother & daughter* and *Foreman* at 48 and 176 kbits/s.

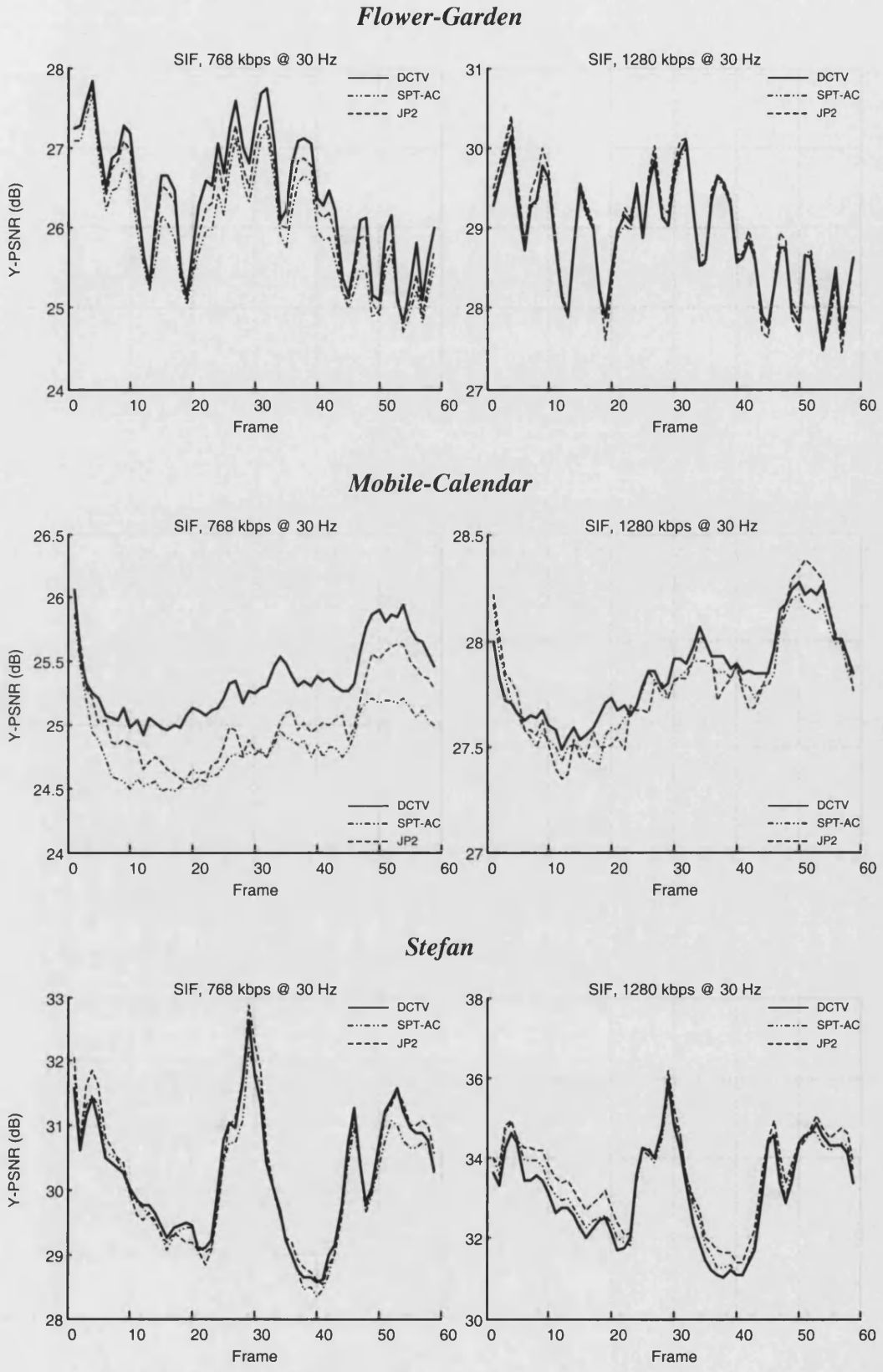


Figure 4.9 Wavelet-based PSNR comparison for *Flower-Garden*, *Mobile-Calendar* and *Stefan* at 768 and 1280 kbits/s.

Figure 4.10 *Akiyo* frame 10 at 128 kbits/s @ 25 Hz by MPEG (left) and the proposed encoder using variable DCT.

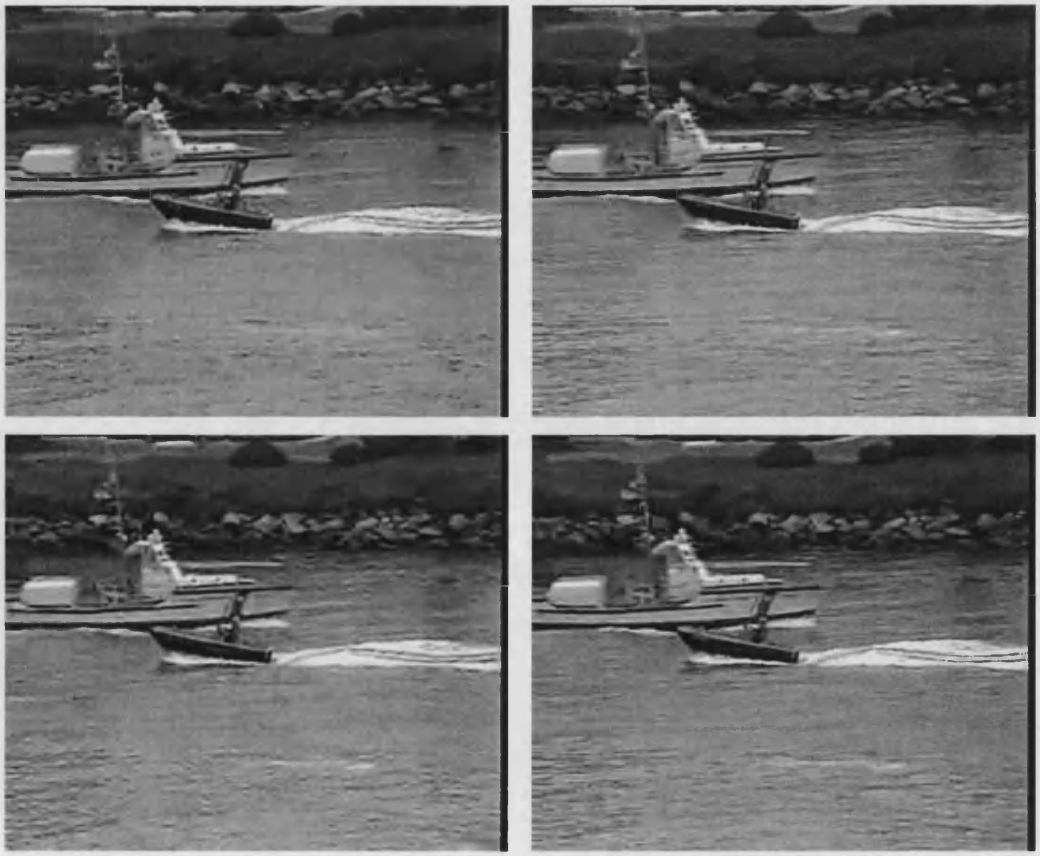


Figure 4.11 *Coastguard* frame 50 at 176 kbps @ 15 Hz from the wavelet-based JPEG-2000 coder (top-left) and SPIHT-AC coder (bottom-left), and the proposed DCT-based coder with (bottom-right) and without deblocking filter (top-right).

noise ratio in *Coastguard*, *Mother & Daughter* and *Foreman*, and a more moderate 0.1 – 0.5 dB for *Flower-Garden* and *Mobile-Calendar* at 768 kbits/s. The differences between the three coders are modest for *Flower-Garden* and *Mobile-Calendar* at 1280 kbits/s, and *Stefan* at 768 kbits/s. For *Stefan* at 1280 kbits/s however, JP2 achieves the best PSNR performance and outperforms both SPIHT-AC and the proposed DCT-based coder by an average of 0.3 dB and 0.4 dB respectively. The PSNR advantage that the wavelet transform is usually thought to have over the DCT in video coding is shown to reside in the way the coefficients of each are encoded, rather than in the transforms themselves. Figure 4.11 shows a sample frame of *Coastguard* from the three different systems coded at 176 kbits/s. Also shown in the figure is a "deblocked" version from the proposed coder, i.e. a deblocking filter is applied to the reconstructed images. As would be expected, blocking artifacts can be seen in the output from the proposed coder without the deblocking filter. However, when the deblocking filter is used, the visual quality improved significantly and a further 0.0 – 0.2 dB improvement in PSNR is possible depending on the deblocking technique being used.

Table 4.3 lists the averaged luminance PSNR coding results obtained from the proposed DCT-based coder and H.263 TMN 3.0 with optimized off-line rate control and Annex F & I switched on. The H.263 results are not shown in Figure 4.8 and 4.9 because the rate control (i.e. bit allocation) in both systems is performed differently. Other differences between the two systems are the advance DC prediction and overlapped block matching compensation (OBMC) routines. To ensure fair comparison, the first I-picture in H.263 TMN 3.0 is quantized such that similar PSNR to the proposed coder is achieved and the H.263 coder is run iteratively so that the net rates for the inter-frames are as close as possible to those obtained from the proposed coder. From the table, it can be seen that the embedded DCT algorithm outperforms H.263 by 0.05 – 1.26 dB in all but two data point.

Table 4.3 PSNR coding results from H.263 TMN 3.0 and the proposed DCT coder.

Sequences	*Target rates*	*Averaged PSNR*		*Rate difference*	*PSNR difference*
		Proposed	*H.263*	*Proposed-H.263*	*Proposed-H.263*
Coastguard	QCIF, 48 kbps @ 10 Hz	29.79	29.74	-1.81	0.05
	CIF, 176 kbps @ 15 Hz	27.72	27.85	0.26	-0.13
Mother & Daughter	QCIF, 48 kbps @ 10 Hz	38.23	36.97	-1.20	1.26
	CIF, 176 kbps @ 15 Hz	40.07	39.36	-1.78	0.71
Foreman	QCIF, 48 kbps @ 10 Hz	32.79	32.46	-0.31	0.33
	CIF, 176 kbps @ 15 Hz	32.91	32.93	-1.45	-0.02
Flower-Garden	SIF, 768 kbps @ 30 Hz	26.39	25.87	-8.04	0.52
	SIF, 1280 kbps @ 30 Hz	28.87	28.45	-3.05	0.42
Mobile-Calendar	SIF, 768 kbps @ 30 Hz	25.34	24.66	-7.33	0.68
	SIF, 1280 kbps @ 30 Hz	27.84	26.92	-2.54	0.92
Stefan	SIF, 768 kbps @ 30 Hz	30.21	29.62	-8.59	0.59
	SIF, 1280 kbps @ 30 Hz	33.17	32.44	-8.13	0.73

4.8 Summary

In this chapter, we have presented a simple and effective embedded residual coding technique that works well across a broad range of sequences and bit rates. In particular, the coder works equally well with mixed-mode (intra- and inter-mode) motion residuals and outperforms MPEG and H.263 by 0.3 – 2.5 dB and 0.05 – 1.26 dB respectively while retaining the computational efficiency of the DCT. The advantage that wavelet transforms have over the DCT in video coding is less obvious than in still image coding. Based on our objective performance results, we have shown that a DCT based system with improved coding of coefficients nearly always produces better PSNR objective performance than the wavelet-based SPIHT-AC coder, which itself outperforms the Jasper 1.500.4 JPEG 2000 codec in low bit rates coding.

Chapter 5

Improved Coding of Matching Pursuits

5.1 Introduction

Transform coding has long proved to be a simple and effective way of obtaining good compression. The discrete cosine transform (DCT) was incorporated into the first still image coding standard JPEG, as well as all video coding standards [44, 45, 46, 47, 49, 50, 51, 52, 53] that are available today. The introduction of wavelets in the mid-1980s, however, has provided researchers and academics alike with another powerful mathematical tool in many areas of science and engineering, but particularly so for the data compression community. Still image compression systems comprising a wavelet transform with subsequent quantization tailored to the transform structure not only provide superior image quality, both objectively and subjectively, compared to the traditional DCT, but it is also capable of generating a bit-stream that is both SNR and resolution scalable. As would be expected, the incorporation of the wavelet transform in video coding has followed from its success in still image coding and competitive results to existing standards have been reported from numerous authors [13, 40, 56, 69, 70, 72, 112, 115].

In trying to achieve very low bit rates for coded video, however, it becomes necessary to re-examine the efficiency of orthogonal transforms such as the DCT and the DWT for encoding the motion compensated residuals that are generated by

the hybrid system design. When the bit budget is small, orthogonal transforms can require more bits than available to represent the most important structures, which are well localized in space and whose Fourier transform are often well localized at high frequencies, in the motion residuals. This arises mainly from the inability of the transforms to have both time and frequency localization, that is, the ability to clearly identify signal events which manifest during a short time interval and at the same time are concentrated at particular Fourier frequencies. Consider, for example, Figure 5.1 where the time-frequency tilings of several transforms are given. Here the signal consists of a single row of pixels taken from frame 1 of the *Stefan* sequence, and its time-frequency characteristic is given by the Wigner-Ville distribution in part (b). In part (c)–(f), the darker the grey level of each Heisenberg box the larger the amplitude of the corresponding coefficients. Of these tilings, it can be seen clearly that time (e.g. the width of the Heisenberg box) and frequency (e.g. the height of the Heisenberg box) resolutions are roughly constant, i.e. one cannot have arbitrarily fine time and frequency resolutions but can trade one for another. In part (c), the DCT is not well adapted to describing structures that are much smaller (or larger) than its window size while in part (e), the DWT has very poor frequency resolution at high frequencies. The adaptive version of both transforms, namely DCT with varying window sizes in part (d) and the wavelet packets in part (f), achieve a compromise by locating the spikes to a certain degree. However, the frequency resolution in both plots is not ideal, as can be seen from the Wigner-Ville distribution.

To overcome these shortcomings, Mallat and Zhang [74] introduced an algorithm called matching pursuit that decomposes any signal f into a linear expansion of waveforms that belong to a redundant dictionary of functions $D = \{\varphi_\gamma\}$, with $\|\varphi_\gamma\| = 1$ for all γ. The algorithm is a recursive procedure, where a waveform $\varphi_{\gamma n} \in D$ that is best adapted to approximate part of the signal is chosen at each iteration. After m iterations, a matching pursuit decomposes the original vector f into a sum of dictionary elements

$$f = \sum_{n=0}^{m-1} \alpha_n \varphi_{\gamma n} + R^m f, \tag{5.1}$$

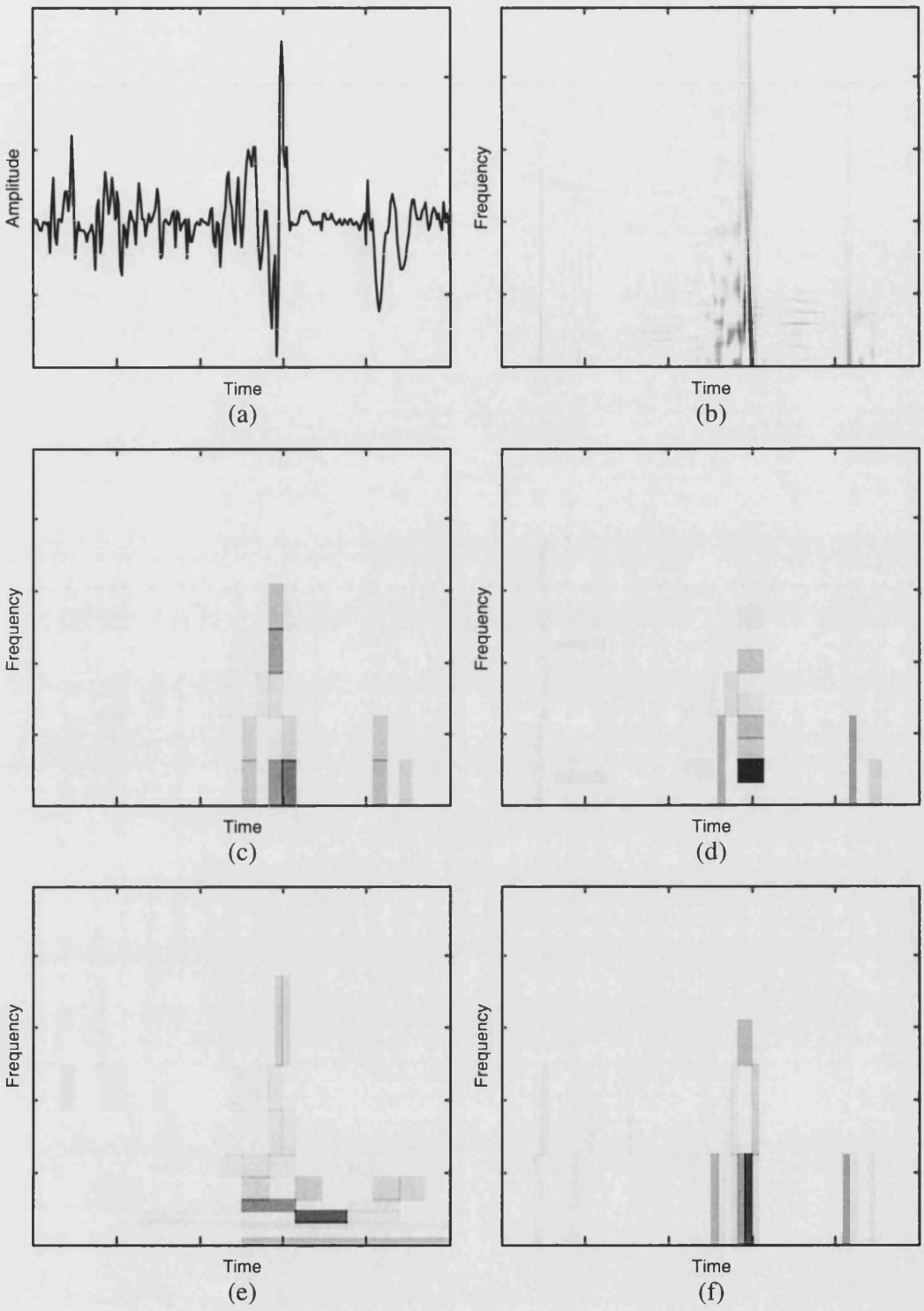


Figure 5.1 Tilings of time-frequency plane. (a) Signal of 352 pixels taken from *Stefan* sequence. (b) Wigner-Ville distribution of the signal shown in (a). (c) Expansion using the local cosine basis with windows of constant size. (d) Expansion using the local cosine basis with windows of varying sizes. (e) Expansion using a discrete wavelet transform. (f) Expansion using the wavelet packets.

where α_n is the matching pursuit coefficient (or inner product) given by

$$\alpha_n = \langle \varphi_{\gamma n}, R^n f \rangle \tag{5.2}$$

and $R^m f$ is the mth order residual vector after approximating f in the direction of $\varphi_{\gamma m-1}$. Since α_n is determined by projection, $\alpha_n \varphi_{\gamma n-1} \perp R^{n+1} f$. Thus we have the "energy conservation" equation

$$\left\| R^n f \right\|^2 = \left\| R^{n+1} f \right\|^2 + \alpha_n^2. \tag{5.3}$$

In practice, however, an exact reconstruction of the signal is rarely possible. First, the number of iterations is generally limited due to computational cost and bandwidth constraints. Thus, the signal is approximated by the first N iterations (e.g. atoms) of the matching pursuit algorithm. Note also that exact signal reconstruction may need an infinite number of iterations with a redundant dictionary. Second, the matching pursuit coefficients and dictionary indices have to be quantized to reduce the overall size of the signal representation. In particular, the matching pursuit coefficient α_n is quantized to $\hat{\alpha}_n = Q(\alpha_n)$ prior to the computation of the residual $R^{n+1} f$, that is

$$R^{n+1} f = R^n f - \hat{\alpha}_n \varphi_{\gamma n}. \tag{5.4}$$

The use of the quantized value in the residual calculation reduces the propagation of the quantization error to subsequent iterations. The reconstruction of the signal is thus given by

$$\hat{f} = \sum_{n=0}^{p-1} \hat{\alpha}_n \varphi_{\gamma n}. \tag{5.5}$$

In this chapter, we describe in detail a video compression system that is based on matching pursuits [78] and show how a bit-plane coder, as used in the previous chapter, can be used to improve its coding efficiency. In particular, the so-called "deadlocking" problem that is associated with the employment of scalar quantizers is addressed through the use of a Precision Limited Quantization scheme.

5.2 Original implementation

Neff and Zahkor [78] were the first to use matching pursuits for encoding motion compensated residuals. The system proposed by the authors is similar to the traditional hybrid motion compensated design depicted in Figure 4.1, except for the residual coder (encoder and decoder) which is based on matching pursuits. Simplified block diagrams for both the encoder and the decoder are shown in Figure 5.2. As can be seen from the figure, the encoder consists of two subcomponents, namely *Find Atoms* and *Code Atoms*. The *Find Atoms* subcomponent decomposes the motion residual into a series of atoms using a computational efficient 2-D separable dictionary that allows fast inner product search. Each atom is defined by a five parameters set $\{\gamma, \psi, x, y, \hat{\alpha}_{\gamma,\psi}\}$ consisting respectively of the best match structure elements from the dictionary, the location of the best match in the motion residual,

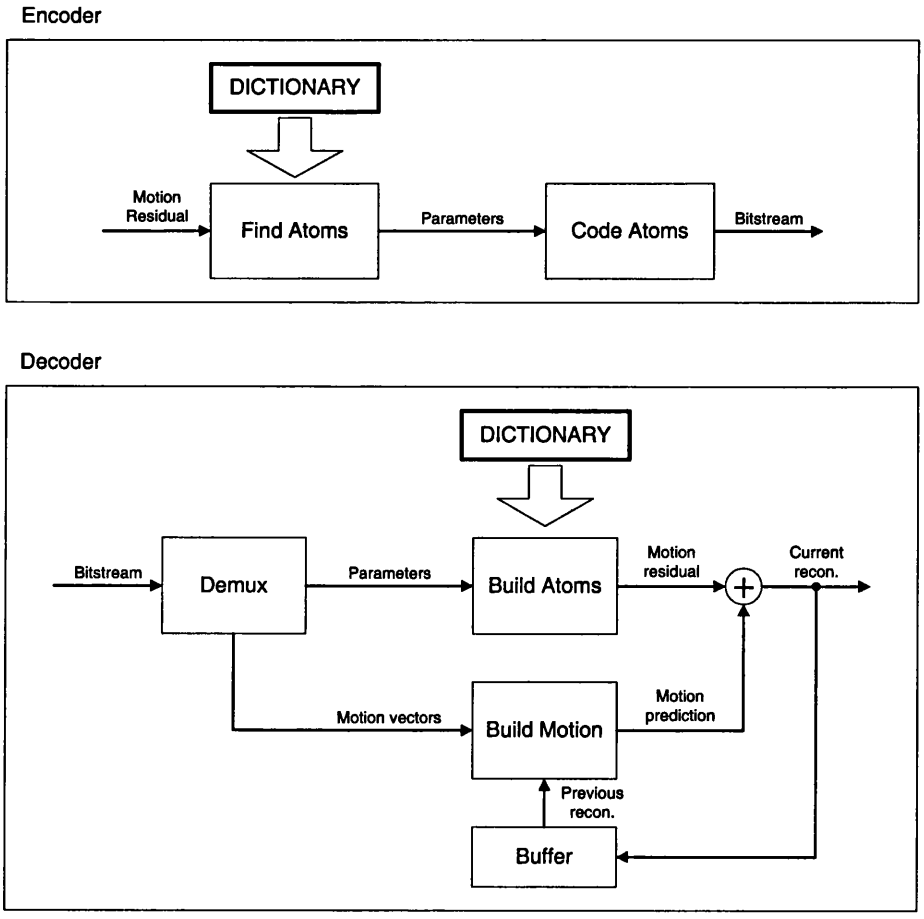


Figure 5.2 Simplified block diagrams for both the encoder and the decoder

and the quantized inner product. The parameters from all the atoms are then Huffman-coded by the *Code Atoms* subcomponent. The decoder reverses the encoder operation by reassembling the atoms using simple lookup tables and reconstructing the motion residual using (5.5).

The dictionary proposed by Neff and Zahkor is made up of 1-D discrete modulated Gaussian windows given by

$$\varphi_\gamma(i) = K_\gamma\, g\left(\frac{i-\frac{N}{2}+1}{s}\right)\cos\left(\frac{2\pi\xi\left(i-\frac{N}{2}+1\right)}{16}+\phi\right),\ i=0,1,...,N-1, \tag{5.6}$$

where the prototype Gaussian window $g(\cdot)$ is defined as

$$g(t) = \sqrt[4]{2}e^{-\pi t^2}. \tag{5.7}$$

Here $\gamma = (s,\xi,\phi)$ is a triple consisting respectively of a positive scale, a modulation frequency and a phase shift. The constant K_γ is chosen such that $\|\varphi_\gamma\| = 1$. The 2-D separable dictionary can therefore be written as

$$\tilde{\varphi}_{\gamma,\psi}(i,j) = \varphi_\gamma(i)\varphi_\psi(j),\quad i,j\in\{0,1,...,N-1\}. \tag{5.8}$$

Figure 5.3 shows a visualization of the 2-D basis set. The dictionary triples and associated sizes that form the 1-D basis set are listed in Table 5.1.

In order to speed up the decomposition process, two simple techniques are used. First, the motion residual is divided into blocks and the energy of each block recorded. The block with the largest energy value is then adopted as an initial estimate for the inner product search. Second, the separability of the dictionary basis set is exploited. Let $S\times S$ be the search window, B be the number of 1-D basis elements used to generate the 2-D set, and L_γ and L_ψ be the associated sizes of the horizontal and vertical basis components respectively. For a given 2-D nonseparable basis element, the inner product can be written as

$$\alpha_{\gamma,\psi} = \sum_{i=0}^{L_\gamma-1}\sum_{j=0}^{L_\psi-1}\tilde{\varphi}_{\gamma,\psi}(i,j)I(i,j), \tag{5.9}$$

where $I(i,j)$ is underlying image patch. The total number of operations required for finding a single atom is thus

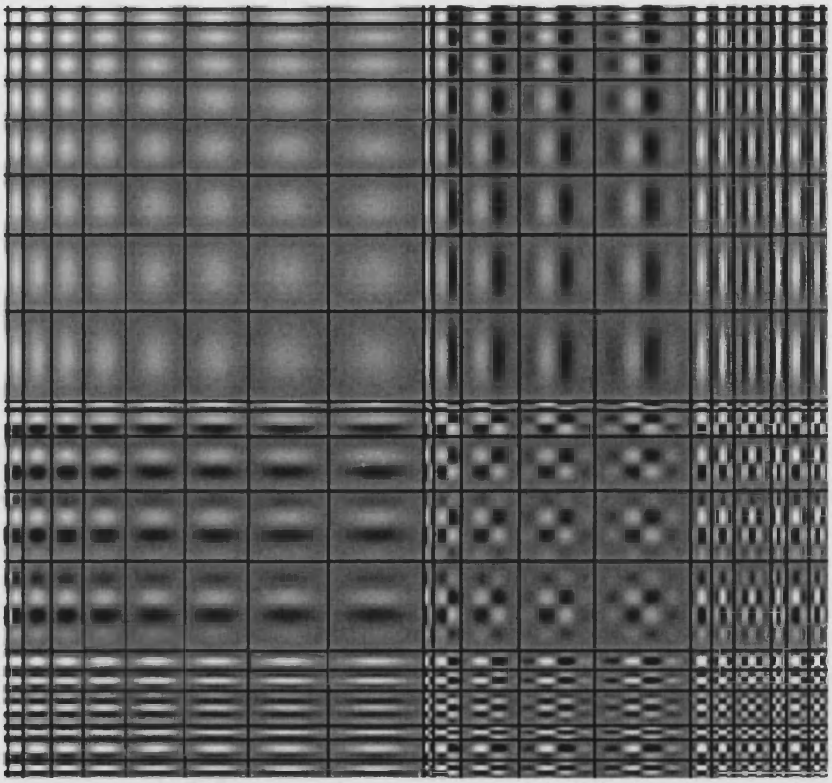


Figure 5.3. 2-D separable Gabor dictionary.

Table 5.1 Dictionary triples and associated sizes that form the 1-D basis set.

k	s_k	ξ_k	ϕ_k	N	k	s_k	ξ_k	ϕ_k	N
0	1.0	0	0	1	10	5.0	1.0	$\pi/2$	9
1	3.0	0	0	5	11	12.0	1.0	$\pi/2$	21
2	5.0	0	0	9	12	16.0	1.0	$\pi/2$	27
3	7.0	0	0	11	13	20.0	1.0	$\pi/2$	35
4	9.0	0	0	15	14	4.0	2.0	0	7
5	12.0	0	0	21	15	4.0	3.0	0	7
6	14.0	0	0	23	16	8.0	3.0	0	13
7	17.0	0	0	29	17	4.0	4.0	0	5
8	20.0	0	0	35	18	4.0	2.0	$\pi/4$	7
9	1.4	1.0	$\pi/2$	3	19	4.0	4.0	$\pi/4$	7

$$T_{nonsep} = S^2 \sum_{\gamma=0}^{B-1} \sum_{\psi=0}^{B-1} L_\gamma L_\psi. \tag{5.10}$$

For S = 16, we get 21.8 million multiply-accumulate operations. If 2-D separable basis element is used, (5.9) becomes

$$\begin{aligned} \alpha_{\gamma,\psi} &= \sum_{i=0}^{L_\gamma - 1} \sum_{j=0}^{L_\psi - 1} \varphi_\gamma(i)\varphi_\psi(j) I(i,j) \\ &= \sum_{i=0}^{L_\gamma - 1} \varphi_\gamma(i) \sum_{j=0}^{L_\psi - 1} \varphi_\psi(j) I(i,j). \end{aligned} \tag{5.11}$$

Computing a single 2-D inner product is thus equivalent to taking L_γ vertical 1-D inner products, each of length L_ψ, and then followed by a single horizontal inner product of length L_γ. The operation count in this case is thus

$$T_{sep} = \sum_{\gamma=0}^{B-1} \left(L_\gamma S(S + L_{\max}) + \sum_{\psi=0}^{B-1} L_\psi \right), \tag{5.12}$$

where $L_{\max}$ is the size of the largest 1-D basis function. A total of 1.7 million multiply-accumulate operations are therefore required for a search window of size 16. Compared to the nonseparable case, this represents a speed improvement of about 13 times.

Having determined the atoms for the motion residual, the last step is to entropy code the five parameter sets $\{\gamma, \psi, x, y, \hat{\alpha}_{\gamma,\psi}\}$ associated with the atoms. In [78], the atoms for each frame are grouped together and coded in position order, from left to right and top to bottom. The positions $\{x,y\}$ are specified with adaptive Huffman codes derived from the previous ten frames worth of position data. Since position data from previous frames is available at the decoder, no additional bits need be sent to describe the adaptation. The three other parameters are coded using fixed Huffman codes. Separate code tables are maintained for the horizontal and vertical shape indices $\{\gamma, \psi\}$. Two special escape code-words in the horizontal luminance shape table are used to indicate when a decoded atom belongs to the C_r or C_b colour difference signals. In all cases, the inner product $\alpha_{\gamma,\psi}$ is quantized by a uniform quantizer with fixed step size, and transmitted using variable length codes.

Neff and coworkers [76, 102] have since made several modifications to the baseline matching pursuit video coder for better coding efficiency and error resilience. In [102], the positions of the atoms are coded on a macroblock basis, where the scanning order for the atoms start at the top left corner and proceed clockwise until the center of the macroblock is reached. The distances between the atoms are coded differentially using four different VLC tables. In the same work, the atoms search strategy is also changed from always using the highest energy block as an initial estimate for the inner product search to a weighted energy search, where the set of weights are linked to the number of times the block has been visited. In [76], they address the quantizer design of the baseline coder, which imposes a hard limit on both rate and quality. That is, the baseline matching pursuit video coder will operate only at rates low enough such that all inner products $\{\alpha_{\gamma,\psi}\}_{\gamma,\psi\in B}$ satisfy

$$|\alpha_{\gamma,\psi}| \geq \frac{QP}{2}, \tag{5.13}$$

where QP is the quantizer step size. This is also known as the "deadlocking" condition. Three methods investigated in [76] are:

1. Modified fixed quantizers design: A simple extension to the fixed quantizer scheme where four quantizers, whose step sizes are logarithmically spaced, are used.
2. Optimum two-pass design: An analysis pass is first used to find the optimum QP and the smallest inner product $\alpha_{\min}$ for a given bit rate. Both parameters are then used in the second quantized pass.
3. Fast one-pass design: Similar to two-pass, but analysis pass is omitted and QP and $\alpha_{\min}$ are predicted from known values in the previous frame.

As would be expected, the optimum two-pass design produces the best ojective performance, then followed by the fast one-pass design and the modified fixed quantizers design. The PSNR improvement attained by both one- and two-pass designs over the modified fixed quantizers design is typically in the order of 0.1 – 0.3 dB in most MPEG-4 test sequences. It must be emphasized that of the three methods proposed, only the optimum two-pass design is guaranteed to avoid the "deadlocking" condition (e.g. (5.13)).

5.3 Precision Limited Quantization

The issue of what distortion measures are more suitable for audio-visual source coding has been the subject of continuing study for the past three decades. Recent advances in technologies and the desire of network users to have access to multimedia content over Public Switched Telephone Networks (PSTN) or mobile channels have accentuated the importance of distortion measures that correlate well with the perceptual impact of the loss. In the case of a "normal" picture, the solution is often based primarily on the concept of spatial and/or frequency masking – the increase in threshold for a given stimulus in the presence of a masking signal [37, 68, 101, 120, 121, 129]. The idea is to construct a perceptual threshold, also known as just-noticeable differences (JND), which attempts to account for the human sensitivity to frequency content (i.e. frequency masking). Very often an additional masking function such as texture masking is used to adjust the perceptual thresholds so that textured regions would not be over-coded.

Precision Limited Quantization (PLQ) is a visually tuned quantization scheme developed by Monro *et al.* [68] primarily for embedded wavelet still image coding. Unlike [37], where the authors only considered the based sensitivity thresholds (as proposed by Safranek [101]) for an embedded coder, the PLQ scheme could be viewed as an advance perceptual model which combines both the base sensitivity thresholds and texture masking function proposed by Safranek [101]. The PLQ quantizes every coefficient in a subband not by the same quantization level, but to the same number of refinement bits R beyond its most significant bit, as illustrated by Figure 5.4(a). It is clear from the figure that for textured regions where coefficients intensities differ greatly, the PLQ exhibits a simple form of texture masking such that coefficients with higher magnitude are quantized more coarsely as compared to those with smaller values in their vicinities. The wavelet subbands are grouped according to their scale, as illustrated in Figure 5.4(b), and the appropriate number of bits to use in each group of subbands was determined by a psychovisual trial involving thirteen subjects. Each image was first coded to its full precision and the subjects were asked to find the refinement depth for each group of subbands, starting from the finest to coarsest, using a mouse driven slider

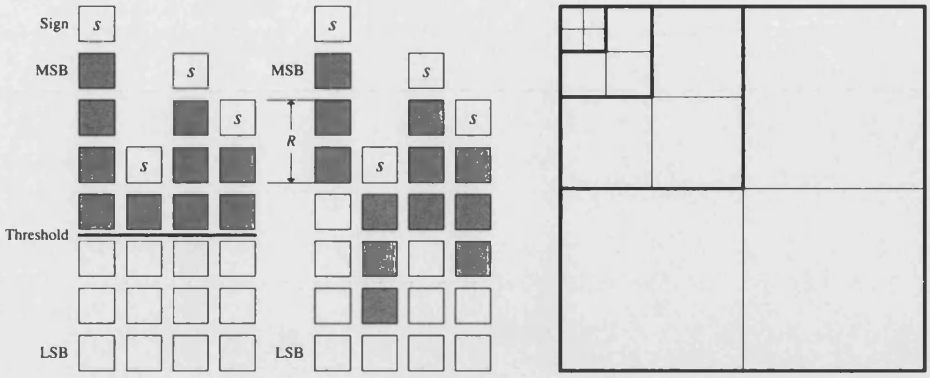


Figure 5.4 Precision Limited Quantization. (a) Scalar quantization to fixed threshold (on the left) and fixed depth. (b) Subbands grouping for determining the refinement depths.

at which they could see no difference. Figure 5.5(a) shows the 95% confidence interval for the refinement depths of three test images (*Barbara*, *Boats* and *Goldhill*) using a 5-scale DWT with the 5/3 biorthogonal filters kernel. A conservative choice for the refinement depths of the three images would be {2,2,3,3,5}, starting at scale 1 from the left to scale 5 at the right. Note that the quantization errors produced by using the refinement depths depicted in Figure 5.5(a) yield just-noticeable differences, i.e. they are visually lossless as compared to the original image. Figure 5.5(b)–(d) shows the minimum refinement depths which offered best visual quality at bit rates of 0.1, 0.2, 0.4 and 0.8 bits per pixel. From the figures, it can be seen that the minimum refinement depths for the three test images tend to saturate around the same values, and in this case {1,1,2,2,4}, regardless of the image content and bit rates. To make sure the refinement depths are indeed independent of both quantities, the One-way Analysis of Variance (ANOVA) statistical test was performed on two sets of data; the first set is of the same scale and bit rate from different images (e.g. image dependent test) and the second set from each individual image only (e.g. bit rate dependent test). Table 5.2 lists the *p*-values for each group of data. Note that in order for the group sample means to be deemed statistically significant, the *p*-values have to be smaller than 0.05. The *p*-values from the image dependent test (bottom half table) clearly demonstrate that PLQ is a fairly robust quantization scheme whereby refinement depths for one image are generally appli-

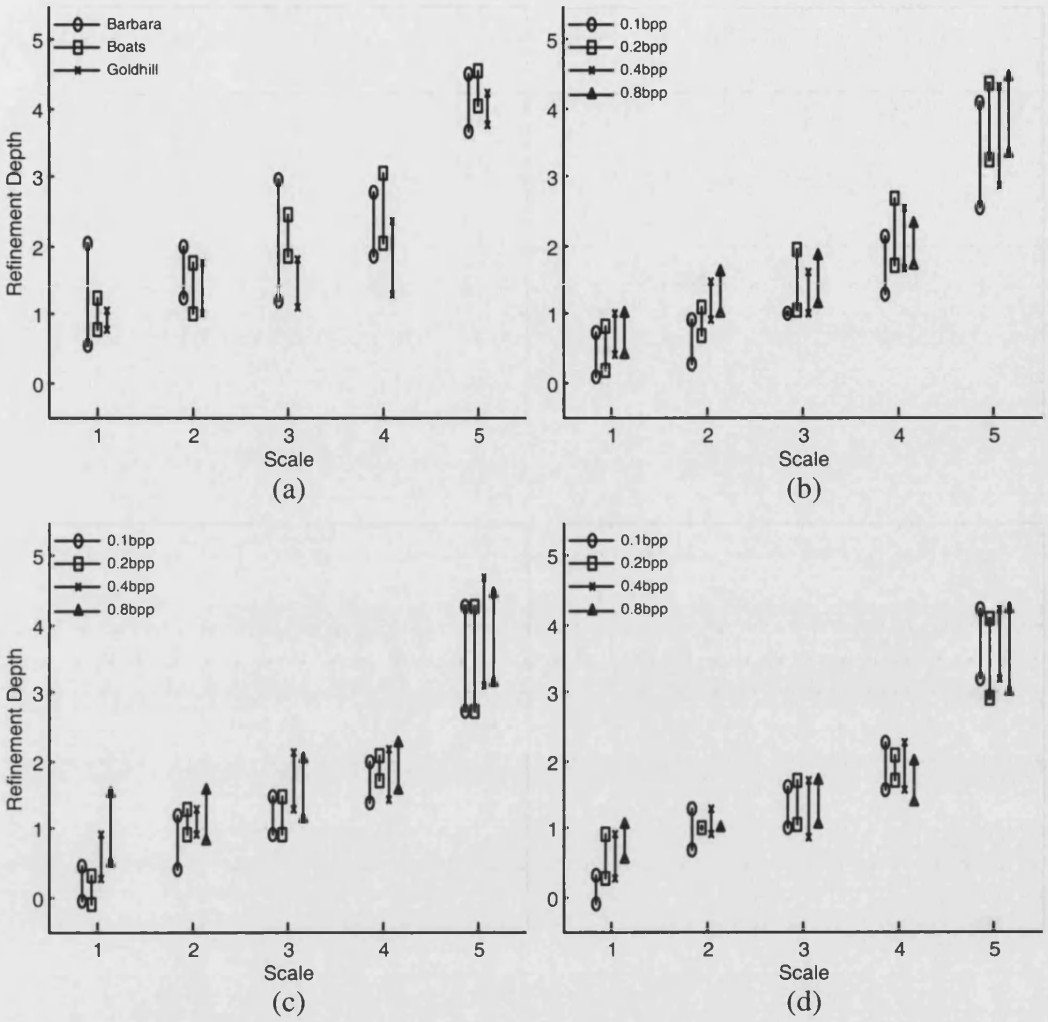


Figure 5.5 95% confidence interval for the refinement depths of three test images using 5-scale DWT. (a) Refinement depths for visually lossless quality. (b)–(d) Refinement depths for *Barbara*, *Boats* and *Goldhill* at selected bit rates.

Table 5.2 *p*-values of ANOVA test for Figure 5.5.

IMG \ SC	1	2	3	4	5
Barbara	0.456	0.005	0.104	0.399	0.610
Boats	0.003	0.324	0.114	0.791	0.835
Goldhill	0.010	0.818	0.954	0.747	0.952

BR \ SC	1	2	3	4	5
0.1	0.293	0.276	0.201	0.677	0.735
0.2	0.054	0.241	0.486	0.349	0.769
0.4	0.878	0.769	0.264	0.582	0.829
0.8	0.530	0.346	0.750	0.418	0.778

cable to other images. The differences in refinement depths between different bit rates are also not significant for all but scale 1 (top half table) of both *Boats* and *Goldhill*.

It is worth noting that the PLQ scheme is not confined to wavelet transform and the approach could be used with coefficients produced by other transforms such as the discrete cosine transform (DCT) and the lapped orthogonal transform (LOT). Of these transforms, an obvious implementation would be to find the refinement depth for each and every subband. While this is feasible in practice, it is very time consuming and it is not clear in what order should the subbands be visited. In the experiments reported here, each 8×8 DCT/LOT block is mapped to a 4 scale dyadic decomposition structure, as depicted in Figure 5.6, so that the procedure that was used on the DWT can now be used on the block transform with only slight modifications. The 95% confidence interval for the refinement depths of visually lossless quality and at bit rates of 0.2 and 0.4 bpp for both the DCT and the LOT are shown in Figure 5.7 and Figure 5.8 respectively. As can be seen from the figures, the refinement depths for both transforms are very similar, e.g. {1,2,2,5}, as is expected since LOT is based on DCT. The image and bit rate dependent tests (see Table 5.3) for the DCT results reveal that PLQ works reasonably well across different images and bit rates. For LOT results however, statistically significant differences can be found at scale 3 and 4 in both tests (see Table 5.4).

For application in video coding, PLQ achieves a depth scheme of all 1s for

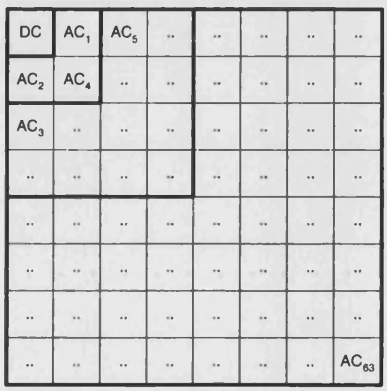


Figure 5.6 Partitioning of 8×8 block transform for PLQ.

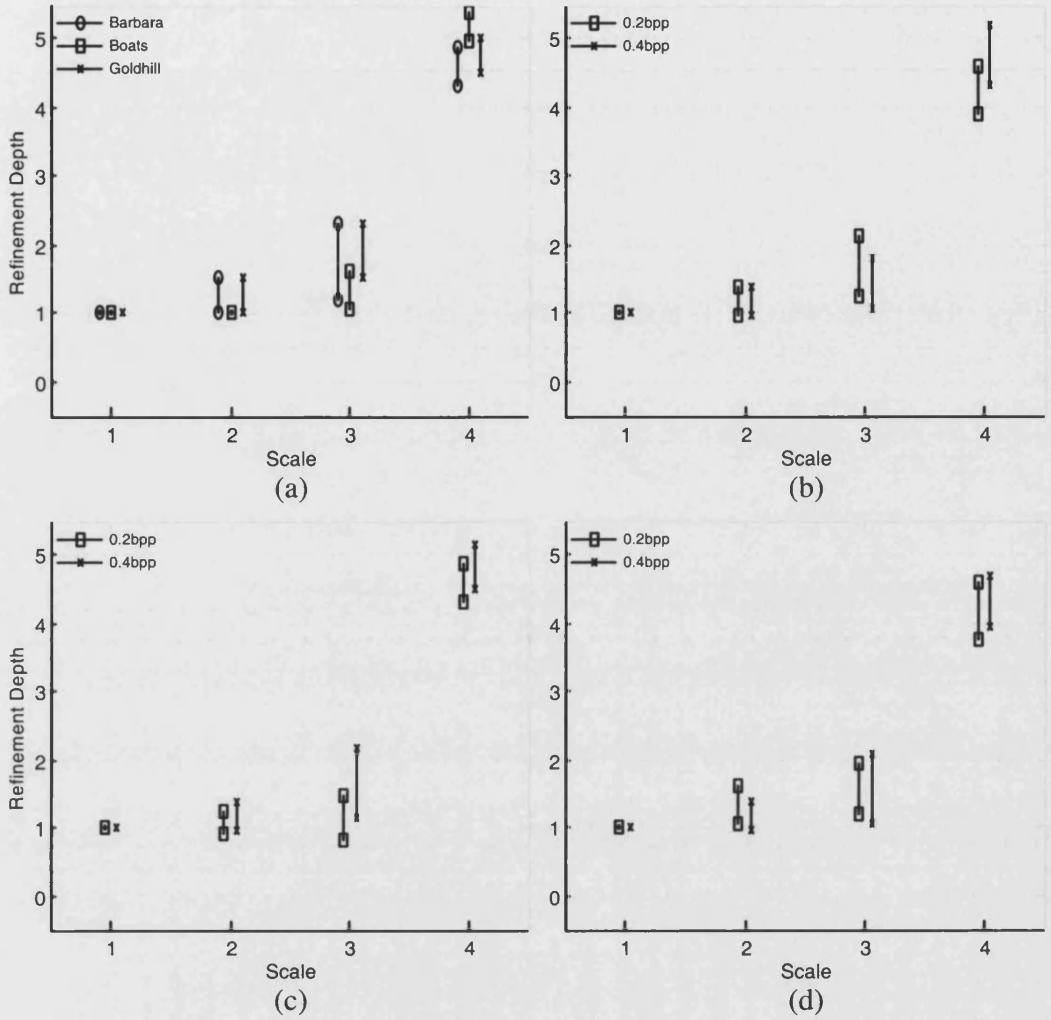


Figure 5.7 95% confidence interval for the refinement depths of three test images using 8×8 DCT. (a) Refinement depths for visually lossless quality. (b)–(d) Refinement depths for *Barbara*, *Boats* and *Goldhill* at selected bit rates.

Table 5.3 *p*-values of ANOVA test for Figure 5.7.

IMG \ SC	1	2	3	4
Barbara	1.000	1.000	0.472	0.096
Boats	1.000	0.635	0.176	0.304
Goldhill	1.000	0.416	1.000	0.558

BR \ SC	1	2	3	4
0.2	1.000	0.410	0.282	0.245
0.4	1.000	1.000	0.797	0.179

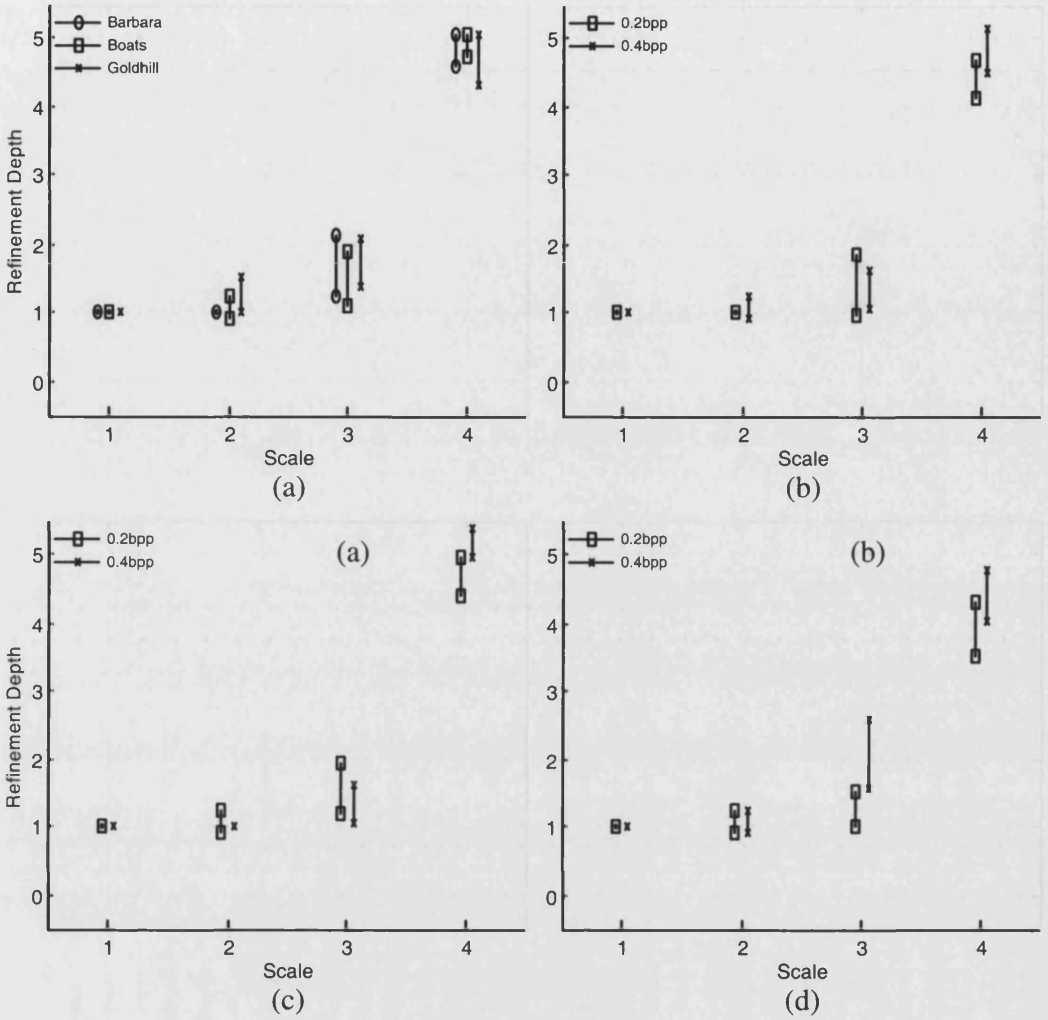


Figure 5.8 95% confidence interval for the refinement depths of three test images using 8×8 LOT. (a) Refinement depths for visually lossless quality. (b)–(d) Refinement depths for *Barbara*, *Boats* and *Goldhill* at selected bit rates.

Table 5.4 *p*-values of ANOVA test for Figure 5.8.

IMG \ SC	1	2	3	4
Barbara	1.000	0.558	0.803	0.090
Boats	1.000	0.558	0.414	0.027
Goldhill	1.000	1.000	0.020	0.081

BR \ SC	1	2	3	4
0.2	1.000	0.808	0.596	0.010
0.4	1.000	0.808	0.034	0.017

all transforms considered so far. Unfortunately, it was found empirically that it has very poor coding efficiency, especially at medium-high bit rates. At these bit rates, the number of macroblocks to be intra-coded in a PLQ-based system increased 5–20 times compared to that of a MSE-based system, and therefore requires more bits to code. This really is no surprise since the motion estimation/compensation processes in all hybrid video compression systems has been designed to use MSE criterion only. Therefore, for PLQ to be employed successfully, one has to make changes not only to the motion model, but also to all bits allocation routines such that human-visual criteria have been addressed.

5.4 Bit-plane Atoms Coding

It is clear from the previous section that every wavelet coefficient w that is quantized by the PLQ scheme can be represented by a triple $Q = \{S,M,R\}$ where S = sign(w), $M = \lfloor \log_2 |w| \rfloor$, e.g. the MSB of $|w|$, R = the refinement bits, an integer in the range $0\text{-}2^{PL-1}\text{-}1$ for $PL > 1$, where PL is the total number of bits that will be used to represent the magnitude. For PLQ to be used in matching pursuits, we replace w by the inner product $\alpha_{\gamma,\psi}$ and include both horizontal and vertical 1-D basis components indices $\{\gamma,\psi\}$ such that $\hat{Q} = \{S,M,R,\gamma,\psi\}$. Thus, given $\hat{Q}$, the quantized inner product is reconstructed as

$$\hat{\alpha}_{\gamma,\psi} = 1 << M + R << (M - PL + 1) + 1 << (M - PL), \tag{5.13}$$

where << denote binary shift and the last term in (5.13) is the result of half-adjusted operation.

To transmit or store the compressed code, we order the five parameters set $\{\hat{Q}_i\}$ by its most significant bit M and its position coded by a modified version of the embedded coder introduced in Chapter 4. Table 5.5 lists the Pseudo code for the modified encoder. Note that the only difference between the two embedded coders is the removal of subsequences s_5 in the new encoder. Also, because the new encoder is not concerned with how the atoms are being gathered, it can be used with

Table 5.5 Pseudo code for position coding of matching pursuit atoms.

- Let n be the index to the bit-plane.
- Let k be the smallest integer satisfying $2^{k+1} > |\alpha_{\gamma,\psi}| \; \forall \; \gamma,\psi \in B$.
- Initialize $\{m_i^n\}$ to zero.

1. $n \leftarrow k$
2. if $n \neq k$
3. Run-length code $\{b_{i,j}^n \mid m_i^{n+1} = 1\}$.
4. Transmit S, R and $\{\gamma,\psi\}$ for $b_{i,j}^n = 1 \; \forall \; i,j$ coded step 3.
5. if $n = 0$
6. Terminate process.
7. end
8. end
9. for $m_i^n \neq 1, i = 1,2,...,N$
10. $m_i^n = 1$ if $b_{i,j}^n \neq 0 \; \forall \; i,j$
11. end
13. Run-length code $m_i^n = 1$ that are found at step 10.
14. Run-length code $\{b_{i,j}^n \mid m_i^n = 1 \text{ at step } 7\}$.
15. Transmit S, R and $\{\gamma,\psi\}$ for $b_{i,j}^n = 1 \; \forall \; i,j$ coded at step 14.
16. $n \leftarrow n$-1

other matching pursuit algorithms that use different dictionaries. The values of S and R are simply transmitted as {0,1} and PL-1 bits of side information respectively, without entropy coding, for each atom as the process signals its position in the motion residual. The dictionary indices $\{\gamma,\psi\}$ are coded using VLC codes, via the same Huffman table in the Neff and Zakhor coder.

It is apparent from Table 5.5 that the embedded coder starts the encoding process from the kth bit-plane to the lowest bit-plane where $n = 0$. This would seem to call into question our assertion that the PLQ scheme does not suffer from the so-

called "deadlocking" condition since it is easy to see that no atoms with $\alpha_{\gamma,\psi} < 1$ will be coded. However, we note that the situation can be remedied easily, simply by pre-scaling the inner products by an arbitrary constant β such that

$$\tilde{\alpha}_{\gamma,\psi} = \beta \alpha_{\gamma,\psi} \tag{5.14}$$

is performed before any actual quantization takes place. If β is selected to be power of 2, then there are few changes in the upper half of the magnitude of the inner products in binary prior to the scaling operation and the pre-scaling operation will has no effect whatsoever on the PSNR performance when the targeted bit rates are small. For the experimental results presented in the next section, $\beta = 32$ is used. Note that with this given β, the smallest inner product that can be coded by the system is 0.03125, which is sufficient for all bit rates of practical relevance.

5.5 Experimental Results

In this section, we compare the performances of transform-based residual coders (e.g. embedded DCT as detailed in previous chapter, 2-scale JPEG-2000 and SPIHT-AC) and the modified MP coder described in Section 5.4 with that of Neff and Zakhor coder [76, 102] which employed macroblock based position coding and modified fixed quantizers design. The performance comparison is carried out on the first 2 seconds of six common test sequences: *Coastguard*, *Mother & Daughter*, *Foreman*, *Flower-Garden*, *Mobile-Calendar* and *Stefan* in either QCIF (176 × 144), or CIF (352 × 288) or SIF (352 × 240) resolution. The first frame for all coders is coded using H.263 DCT intra mode and the remaining frames coded by the respective coders. Coding was performed on the luminance component only. To synchronize the bit rate between the systems, the Neff and Zakhor encoder is first run using a target bit rate and frame rate. A record is kept of the number of bits used to represent each frame. The transform-based and modified MP encoders then use this record to encode the same set of original frames with the same number of bits for each frame.

For easy interpretation, the coding results are separated into two figures. Inter-frame PSNR results (averaged over all frames) from the MP coder by Neff and coworkers and the transform-based embedded coders are shown in Figure 5.9. As would be expected, the matching pursuits system outperforms all transform-based methods typically by the order of 0.1 – 1.0 dB in peak signal-to-noise ratio in all but *Foreman* at 24 kbps and the *Flower-Garden* sequence. In Figure 5.10, coding results from both MP coders – one original and the other one as described in the previous section – are plotted as differential values, measured with respect to the Neff and coworkers coder. Coding results with *PL* set to {1,2,3} are shown. As can be seen from the figure, choosing the best precision limit will always make the proposed method better than the Neff and coworkers coder except in one case where it is virtually the same and another where it is 0.1 dB worse. The best precision limits found are perhaps surprising. At low bit rates the best choice is often 2 bits in total, i.e. the MSB plus one refinement bit. At high bit rates the best choice is consistently 1 bit in total, i.e. only the MSB and sign are coded for each atom.

In Figure 5.11, the atoms count differences for the various PLQ-based MP coders, measured with respect to Neff MP coder, were plotted. As can be seen from the figure, the new quantization scheme allows a very large amount of extra atoms to be coded for *Flower-Garden*, *Mobile-Calendar* and *Stefan*, which may explain why the PSNR is better. However, for the remaining sequences, the situation is largely the opposite. *Mother & Daughter*, in particular, registered the most losses, but amazingly, posted the most significant performance gain of all sequences. One possible explanation for this performance is this: during a closer inspection of the atoms statistics, it was found that this sequence has an unusually high amount of atoms with its inner products just below the quantization thresholds, and therefore suffered from more quantization noise as compared to the modified coder.

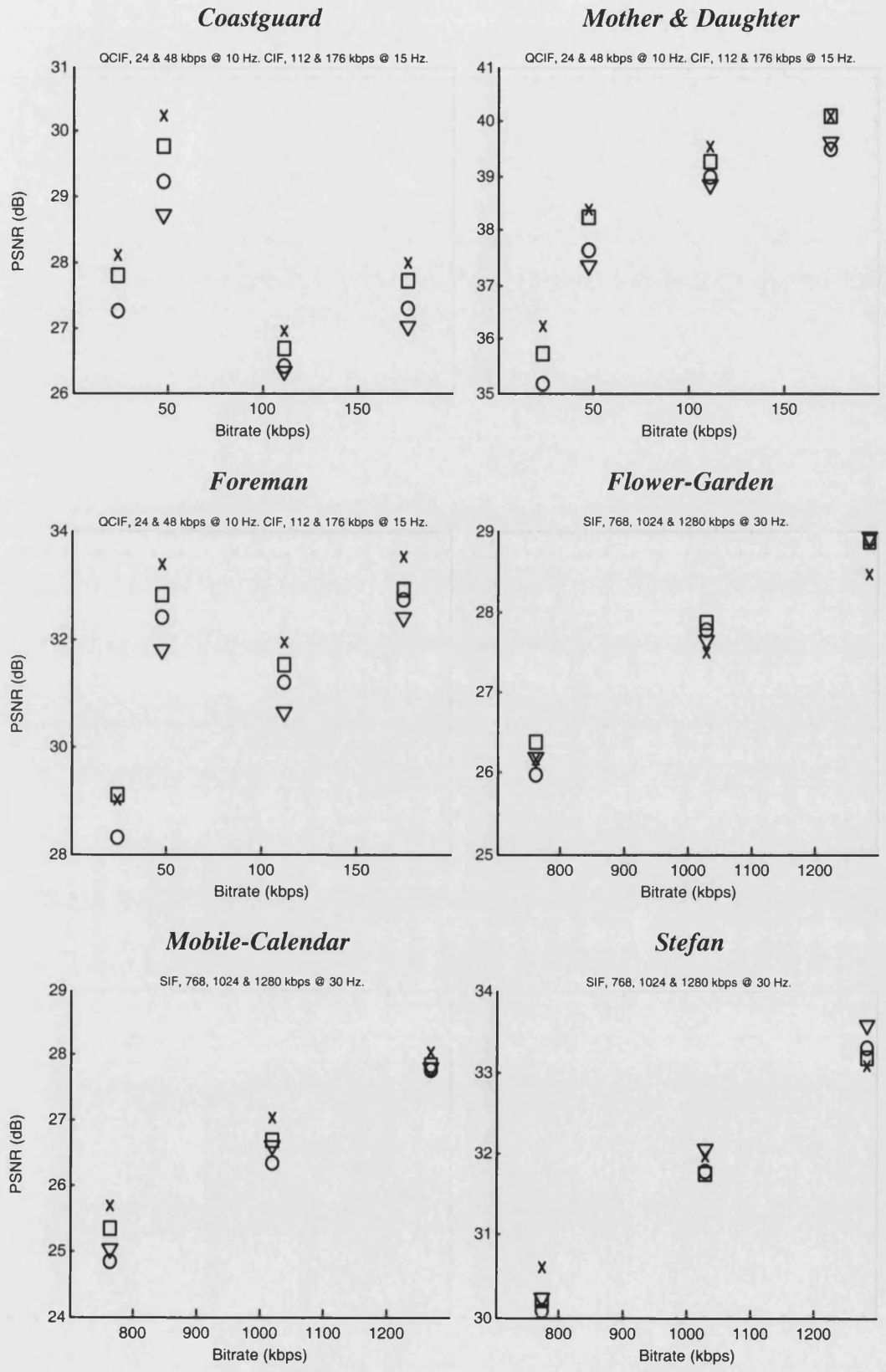


Figure 5.9 PSNR comparison for Embedded DCT (□), 2-scale JPEG-2000 (▽), Neff's MP coder (×) and SPIHT-AC (○) for six common test sequences.

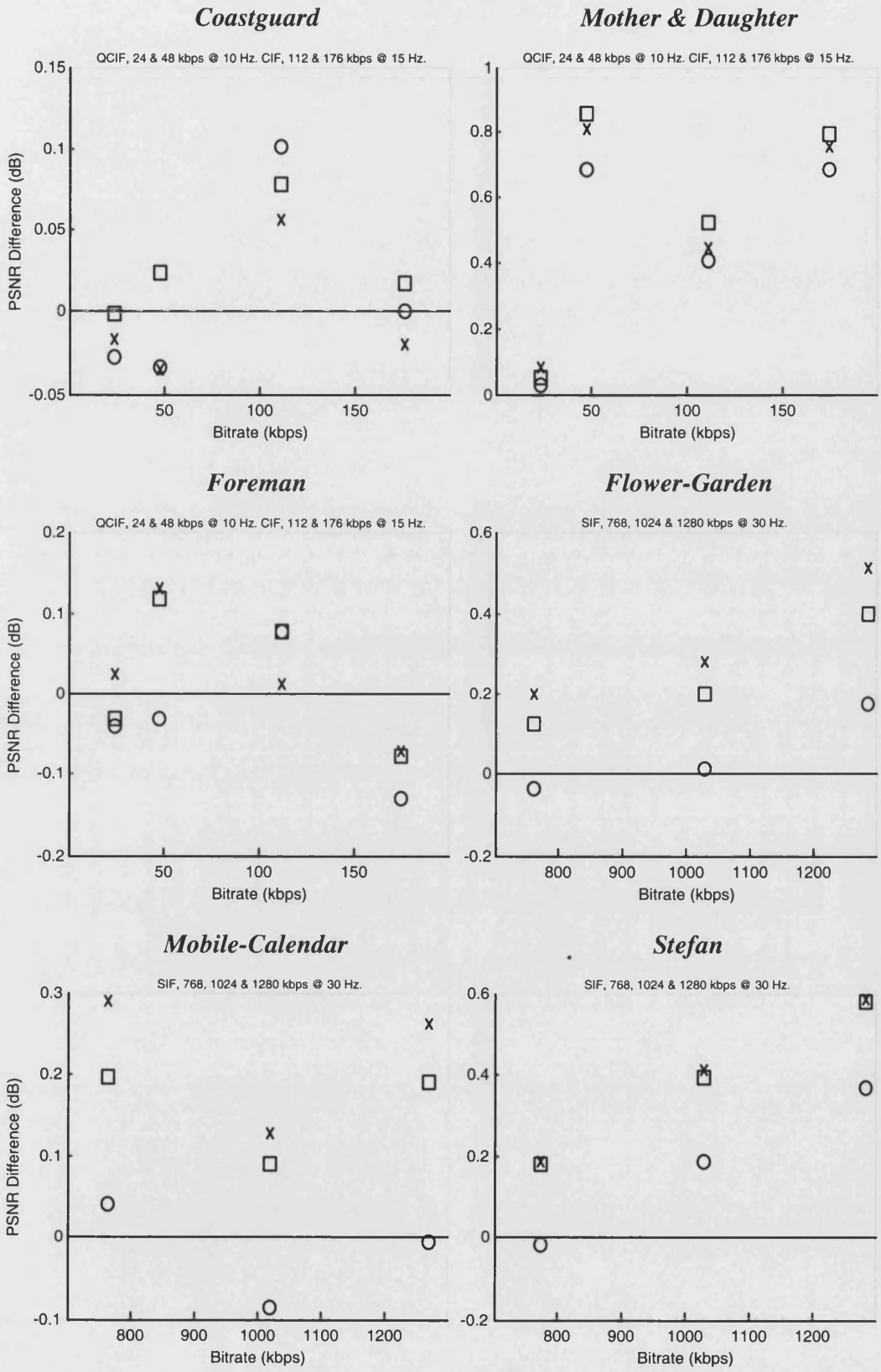


Figure 5.10 PSNR performance gap of PLQ-based embedded coder with $PL = 1$ (×), 2 (□) and 3 (○), measured with respect to the PSNR of Neff and Zakhor coder.

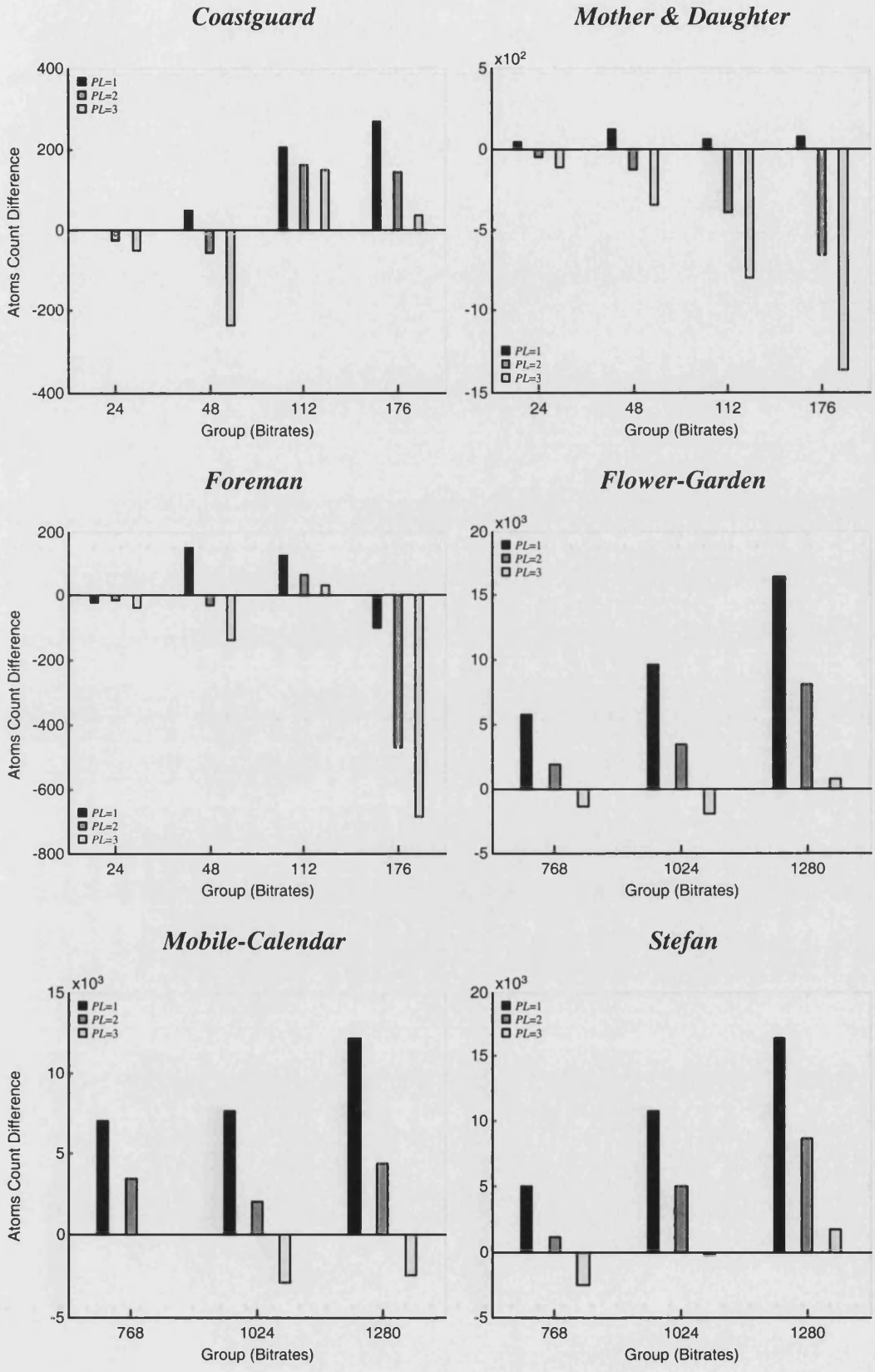


Figure 5.11 Atoms count differences for the PLQ-based embedded coders depicted in Figure 5.10, measured with respect to Neff and Zakhor coder.

5.6 Summary

In this chapter, we have presented a simple and effective embedded matching pursuits coding technique that outperforms the Neff and Zakhor coder in all but one sequence coded at bit rates between 24 – 1280 kbps. In particular, the modified coder does not suffer from the co-called "deadlocking" effect as seen in Neff's works. The performance gain achieved by the modified coder is also fairly significant, ranging from 0.0 – 0.9 dB higher in most test cases. Also, the simple run-length encoder used in the experiments is capable of further improvement. The block masking method used could be seen to provide a simple form of context adaptation. However, the use of a more sophisticated context model for the run length coder used to position atoms, with or without a block mask, would be expected to give further improvement.

Chapter 6

Conclusions

6.1 Summary

The purpose of this thesis was to investigate the coding efficiency of the discrete wavelet transform on motion compensated residuals, as compared to the popular discrete cosine transform. The objective was to create an efficient motion residual compression technique that produces a so-called embedded bit-stream which has the property that the prefixes of the bit-stream yield a continuum of lower rate description of the image at the highest possible levels of quality.

Chapter 1 described the motivation for this project and provided a literature review on the works that have been done so far. In the case of wavelet video coding, the current development generally falls into three categories: 1) 3-D subband coding, 2) motion compensation in the wavelet domain and 3) the replacement of the discrete cosine transform by the discrete wavelet transform in coding motion residuals from the traditional time domain motion estimation and compensation process. Of the three categories, method 1 and 2 offer the possibility of incorporating SNR, temporal and resolution scalability in a single bit-stream. Apart from the wavelet transform, another important development in video coding is the expansion using an overcomplete basis termed matching pursuits. Unlike transform or subband coding, the overcompete basis can contain a wider variety of structures, most

of which offer better time-frequency localization as compared to 2-D linear transformation.

Several video coding standards, as well as their applications, were also described in detail in Chapter 1. The key to achieving a substantial compression ratio is to exploit both the spatial and temporal redundancies within the video sequences. Spatial redundancy is eliminated by using a block-based discrete cosine transform (DCT) coding of 8×8 pixel blocks followed by scalar quantization, zigzag scan, and variable length coding of the quantized indices. Temporal redundancy is exploited by using a block-based motion-tracking technique known as motion estimation and compensation. One exception to these is the H.264 standard – the latest standard to emerge from ISO and ITU. In H.264, the popular 2-D discrete cosine transform has been replaced by a close approximation which enables 16-bit integers arithmetic (for 8-bit sources) to be used throughout the encoding process. The new transform also eliminates the mismatch problem at the decoder.

Chapter 2 provided a mathematical background of filter banks theory and its relation to the wavelet transform. Filter banks are efficient convolution structures that have been extensively used in subband coders for speech, and in transmultiplexers (devices to covert time-division multiplexed data to frequency-division multiplexed data and vice-versa). Wavelet transform, on the other hand, is a more recent advance that involved representing general functions in terms of simpler, fixed building blocks at different scales and position, much as the Fourier transform represents signals in terms of elementary periodic waves. The fact that filter banks can be used to approximate (in theory, scaling or wavelet function is obtained after infinite iterations whereas the filter banks, in practice, seldom goes beyond 6 levels of decompositions) wavelet transform has accelerated its uses in a wide range of applications. Unique to wavelet transform is the necessity to deal with the finiteness of the input signals. Two techniques that solved the boundary problem are symmetric extension and boundary filters. Of these, symmetric extension has been preferred in most applications because of its simplicity, but it requires the filters to have linear-phase.

Chapter 3 compared the objective performance of several linear transforms on motion residuals coding. To better appreciate theirs significance, motion residuals coding with no transform was also presented. Statistical modeling using Generalized Laplacian Distribution was used to obtain the theoretical rate-distortion performance bounds for the various transforms considered. Two scalar quantization schemes that are found in most video coding standards were also used as a basic framework in investigating the operational rate-distortion performance bounds. For the motion residuals considered in the chapter, the employment of linear transforms in a video compression system was shown to provide 0.5 – 1.0 dB improvement in peak signal-to-noise ratio. In the case of discrete wavelet transform (using 9/7, 5/3 or haar filters), 2-levels of decomposition were found to be sufficient for motion residual coding as additional levels of decomposition provide little, if any, performance gain. Compared to the popular discrete cosine transform, the objective performance of both transforms is comparable. Further experiment involving the adaptive version of both transforms indicated that there is no clear winner between the two.

Chapter 4 presented a simple and effective embedded motion compensated residual coding technique that uses the discrete cosine transform with varying window size. The window size to use for each macroblock is determined by a fast hard-thresholding method. Once the window size is determined, the encoding of each bit-plane is driven by a significance block map that is created to take advantage of the clustering of large (or small) variance coefficients inherent in motion compensated residuals. A simple classification scheme based on the significance block map is then applied to the identified significant coefficients to improve coding efficiency. Experimental results have shown that the proposed method outperforms MPEG-2 and H.263 in coding of inter-frames by 0.3 – 2.5 dB and 0.05 – 1.26 dB respectively. In most cases it also outperforms JPEG-2000 and SPIHT-AC and has lower complexity. Blocking artifacts are also less visible in some sequences coded at low bit rates.

In Chapter 5, the embedded coding method developed in Chapter 4 was applied to a video coding technique using matching pursuits. Matching pursuits,

unlike 2-D linear transforms, is a greedy algorithm that uses successive approximation techniques with an overcomplete dictionary of prototype waveforms on the motion residuals. In its original implementation, the algorithm suffers from a rate/quality limiting condition known as "deadlocking". The new embedded coding method, together with Precision Limited Quantization – a visually tuned quantization scheme from the author's earlier work – alleviates this problem and improved upon the original coder by 0.0 – 0.9 dB in all but one sequence coded at bit rates between 24 – 1280 kbps. Since the proposed method places no restriction on the dictionary used, it can be used with any other dictionary optimized either for speed or quality.

In conclusion, the objectives of this thesis have been met. The simple DCT-based embedded coding technique has been shown to outperform MPEG-2, H.263 and some selected wavelet-based coders under consideration, averaging 0.3 – 2.5 dB in most test cases. The embedded coding technique, together with a carefully chosen quantization scheme, has also been shown to improve upon the current state-of-the-art video coding technique based on matching pursuits.

6.2 Future Work

Improvements in embedded coding technique

Time constraint has prevented several enhancements/additions to be made to the embedded coding technique developed in Chapter 4. Key areas to be investigated include:

- *Completeness*: The embedded coding technique has been designed and tested using mainly the luminance components from the test sequences. For the algorithm to be of any practical value, the chrominance components C_r and C_b have to be included as well. A preliminary result involving the chrominance components, where each component is coded independently, has been shown to outperform MPEG-2 by as much as 1.0 dB for some sequences coded at a bit rate

of 2.5 Mbps. However, it is anticipated that the coding performance may be improved by combining all colour components as a single unit.

- *Advanced context modeling*: The technique introduced in Chapter 4 has a decided bias; this was partially the result of a conscious effort to maintain low-complexity. By incorporating a more sophisticated context model for the run length coder, improved performance can be expected.

- *Multiple cosine bases*: The embedded coding technique uses only two bases (e.g. 4×4 and 8×8) so that side information is kept to a minimum. A better alternative, where multiple cosine bases can be used without any penalty, is to transfer the decision making process to the motion estimation stage. This can be achieved by simply using the latest motion estimation algorithm from the H.264 standard.

Wavelet video coding

The future of wavelet video coding is by no means "dead". Key areas to be investigated include:

- *Filter kernel*: The performance differences between the discrete wavelet transform and the discrete cosine transform on motion residual coding has been shown to be marginal in Chapter 3. However, the study has concentrated only on filters having compact support, i.e. filters with finite nonzero coefficients, and generally possessing high regularity. Since motion residuals characteristics are anything but smooth, it is conceivable that by relaxing the number of zeros on the Nyquist frequency, the extra freedom can be used to optimize the filter specifically for video coding. Apart from filters with compact support, wavelet theory also includes filters with infinite nonzero coefficients [43, 84], e.g. IIR filters, and the so-called "multiwavelets" [105], which can possess both linear-phase and orthogonal properties simultaneously. Both types of filters offer new possibilities in video coding. More importantly, Marpe *et al.* have already dem-

onstrated in [73] a wavelet-based video coding solution using IIR filters that outperformed the latest H.264 standard.

- *Scalability*: The multiresolution approach in wavelet transform provides a convenient structure whereby SNR, temporal and resolution scalability can be constructed into a single bit-stream; this is a big advantage compared to the current coding standards. Yang and Ramchandran [136] and Taubman [109] provided some ideas on such a system. The main obstacle with Yang's approach is that subband decomposition is a space variant process, that is, translational motion between two consecutive frames may not be translated into a translational motion between two consecutive wavelet transformed frames. Also, when motion estimation is carried out in the wavelet domain, it is not clear how intra-mode macroblocks, which can be very useful for uncovered areas, can be implemented. This same difficulty also arises in Taubman's work that uses 3D subband transformation.

Matching pursuits video coding

- *Speed*: Matching pursuits is by far the most impressive video compression algorithm to-date (note that direct comparison between matching pursuits and H.264 has however, not yet been made). Its main advantage resides in very low bit rates applications where the visual quality from the compression system does not suffer from either blocking or ringing artifacts. However, in order for it to be viewed on an equal footing with DCT- or wavelet-based systems, its high computational load must be reduced to acceptable level.

Author's Publications

1. D.M. Monro and W. Poh, "Improved coding of atoms in matching pursuits", *to appear in Proceedings of the IEEE International Conf. on Image Proc. 2003.*
2. W. Poh and D. M. Monro, "Comparison of residual compression methods in motion compensated video," *in Proceedings of the IEEE International Workshop on Multimedia Signal Proc.*, Virgin Islands, Dec. 2002.
3. W. Poh and D.M. Monro, "Improved Compression of Motion Compensation Residuals," *in Proceedings of the IEEE International Conf. on Image Proc.*, vol. I, pp. 725-728, Sept. 2002.
4. D.M. Monro, J.L. Aufranc, M.A. Bowers, and W.K. Poh, "Visual embedding of wavelet transform coefficients," *in Proceedings of the IEEE International Conf. on Image Proc.*, pp. 186 - 189, Sept. 2000.

References

1. N. Ahmed, T. Natarajan, and R.K. Rao, "Discrete cosine transform," *IEEE Trans. on Computers*, vol. 23, pp. 88-93, Jan.1974.
2. E. Asbun, P. Salama, E.J. Delp, "Encoding of predictive error frames in rate scalable video codecs using wavelet shrinkage," *in Proceedings of the IEEE International Conf. in Image Proc.*, pp. 832-836, Oct. 1999.
3. E. Asbun, P. Salama, and E.J. Delp, "A rate-distortion approach to wavelet-based encoding of predictive error frames," *in Proceedings of the IEEE International Conf. on Image Proc.*, vol. 3, pp. 150-153, Sept. 2000.
4. J. Barbarien, F. Ansorge, P. Schelkens, and J. Cornelis, "A wavelet-based video coding scheme using mesh-based motion estimation," *in Proceedings of International Workshop on Very Low Bitrate Video Coding*, 2001.
5. M. Banham and J. Brailean, "A selective update approach to matching pursuits video coding," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 7, pp. 119-129, Feb 1997.
6. D. Blasiak and W.Y. Chan, "Efficient wavelet coding of motion compensated prediction residuals," *in Proceedings of the IEEE International Conf. on Image Proc.*, vol. 2, pp. 287-291, 1998.
7. F. Bellifemine, A. Capellino, A. Chimienti, R. Picco, and R. Ponti, "Statistical analysis of the 2D-DCT coefficients of the differential signal for

images," *Signal Processing: Image Communication*, vol. 44, pp. 477-488, Nov. 1992.

8. T. Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*, Prentice-Hall, Englewood Cliffs, 1971.

9. T. Berger, "Optimum quantizers and permutation codes", *IEEE Trans. on Information Theory*, vol. 18, pp. 759-765, Nov. 1972.

10. T. Berger, "Minimum entropy quantizers and permutation codes," *IEEE Trans. on Information Theory*, vol. 28, pp. 149-157, Mar. 1982.

11. K.A. Birney and T.R. Fischer, "On the modeling of DCT and subband image data for compression," *IEEE Trans. on Image Processing*, vol. 4, pp. 186-193, Feb. 1995.

12. R. E. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. on Information Theory*, vol. 18, pp. 460-473, July 1972.

13. G. Bhutani and W.A. Pearlman, "Image sequence coding using the zero-tree method," *in Proceedings of SPIE Conference Visual Communications and Image Proc.*, vol. 2094, pp. 463-471, 1993.

14. I. Balasingham and T. A. Ramstad, "On the relevance of the regularity constraint in subband image coding," *in Proceedings of Asilomar Conference on Signals, Systems, and Computers*, 1997.

15. C.M. Brislawn, "Classification of nonexpansive symmetric extension transforms for multirate filter banks," *Applied and Comp. Harmonic Anal.*, vol. 3, pp. 337-357, 1996.

16. P. Czerepinski, C. Davies, N. Canagarajah, and D. Bull, "Matching pursuits video coding: Dictionaries and fast implementation," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 10, no. 7, pp. 1103-1115, Oct. 2000.

17. Y.T. Chou, W.L. Hwang, and C.L. Huang, "Very low bit rate video coding based on gain-shape VQ amd matching pursuits," *in Proceedings of the IEEE International Conf. on Image Proc.*, pp.76-80, Oct. 1999.

18. G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video Coding at Low Bit Rates," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 8, no. 7, pp. 849-866, Nov. 1998.

19. P.C. Cosman, R.M. Gray, and M. Vetterli, "Vector quantization of image subbands: A survey," *IEEE Trans. on Image Processing*, vol. 5, no. 2, pp. 202-225, Feb. 1996.

20. M. Crouse and K. Ramchandran, "Joint thresholding and quantizer selection for transform image coding: entropy-constrained analysis and applications to baseline JPEG," *IEEE Trans. on Image Processing*, vol. 6, no.2, pp. 285-297, Feb. 1997.

21. P.A. Chou, T. Lookabaugh, and R.M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. on Information Theory*, vol. 35, pp. 299-315, Mar. 1986.

22. B.B. Chai, J. Vass, and X. Zhuang, "Significance-linked connected component analysis for wavelet image coding," *IEEE Trans. on Image Processing*, vol. 8, no. 6, pp. 774-784, Jun. 1999.

23. I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. on Pure and Appl. Math.*, vol. 41, pp. 909-996, Nov. 1988.

24. I. Daubechies, "Orthonormal bases of compactly supported wavelets II, Variations on a theme," *SIAM J. Math. Anal.*, vol. 24, pp. 499-519, Mar. 1993.

25. I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, pp. 247-269, 1998.

26. T.R. Fischer, "A pyramid vector quantizer," *IEEE Trans. on Information Theory*, vol. 32, pp. 568-583, Jul. 1986.

27. N. Farvardin and J.W. Modestino, "Optimum quantizer performance for a class of non-Gaussian memoryless sources," *IEEE Trans. on Information Theory*, vol. 30, pp. 485-497, May 1984.

28. G.D. Forney, "The viterbi algorithm," *in Proceedings of the IEEE*, vol. 61, no. 3, pp. 268-278, Mar. 1973.

29. A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. on Information Theory*, vol. 25, pp. 373-380, Jul. 1979.

30. A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Norwell, MA:Kluwer Academic Press, 1990.

31. S.W. Golomb, "Run-length encodings," *IEEE Trans. on Information Theory*, vol. 12, pp. 399-401, Jul. 1966.

32. R.M. Gray, *Source Coding Theory*, Kluwer Academic Press, 1990.

33. R.G. Gallager and D.C. Voorhis, "Optimal source codes for geometrically distributed integer alphabets," *IEEE Trans. on Information Theory*, vol. 21, pp. 228-230, Mar. 1975.

34. H.M. Hang and J.J. Chen, "Source model for transform video coding and its application – part I: Fundamental theory," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 7, pp. 287-298, Apr. 1997.

35. Y. Huang, H.M. Dreizen, and N.P. Galatsanos, "Prioritized DCT for compression and progressive transmission of images," *IEEE Trans. on Image Processing*, vol. 1, pp. 477-487, Oct. 1992.

36. C. Herley, "Boundary filters for finite-length signals and time-varying filter bank," *IEEE Trans. on Circuits and Systems II*, vol. 42, pp. 102-114, Feb. 1995.

37. I. Hontsch, L. J. Karam, and R. J. Safranek, "A perceptually tuned embedded zerotree image coder," *in Proceedings of IEEE International Conf. on Image Proc.*, pp. 41-44, Oct. 1997.

38. E.S. Hong and R.E. Ladner, "Group testing for image compression," *in Proceedings of the Data Compression Conference*, pp. 3-12, Mar. 2000.

39. Z. He and S.K. Mitra, "A Unified rate-distortion analysis framework for transform coding," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 11, no. 12, pp. 1221-1236, Dec. 2001.

40. G. Heising, D. Marpe, H.L. Cycon, and A.P. Petukhov, "Wavelet-based Very Low Bit-Rate Video Coding Using Image Warping and Overlapped Block Motion Compensation," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 148, no. 2, pp. 93-101, Apr. 2001.

41. B.G. Haskell, A.Puri, and A.N. Netravali, *Digital video: An introduction to MPEG-2*, Chapman & Hall, 1997.

42. J.-Y. Huang and P.M. Schultheiss, "Block quantization of correlated Gaussian variables," *IEEE Trans. Communication*, vol. 11, pp. 289-296, Sept. 1963.

43. C. Herley and M. Vetterli, "Wavelets and recursive filter banks," *IEEE Trans. on Signal Processing*, vol. 41, no. 8, pp. 2536-2556, Aug. 1993.

44. ISO/IEC IS 10918, "Digital Compression and Coding of Continuous-tone Still Images," 1991.

45. ISO/IEC 11172, "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbps," 1993.

46. ISO/IEC 13818-2, ITU-T Recommendation H.262, "Generic coding of moving pictures and associated audio, Part 2: Video," 1994.

47. ISO/IEC 14496-2, "Coding of audio-visual objects - Part 2: Visual," 1999.

48. ISO/IEC FCD15444-1, "JPEG2000 Final committee draft version 1.0," Mar. 2000.

49. JVT: ISO/IEC and ITU-T, "Committee Draft of Joint Video Specification (ITU-T Rec. H.264 / ISO/IEC 14496-10 AVC)," Doc. JVT-C167, May 2002.

50. ISO/IEC FDIS 14496-2, "Generic Coding of audio-visual objects: (MPEG-4 video), Final Draft International Standard," Document N2502, 1999.

51. ITU-T Recommendation H.261, "CODEC for audio-video services at p x 64 Kbps," 1993.

52. ITU-T Recommendation H.263 Version 1, "Video coding for low bitrate communication," Mar. 1996.

53. ITU-T Recommendation H.263 Version 2, "Video coding for low bitrate communication," Jan. 1998.

54. R.L. Joshi, H. Jafarkani, J. H. Kasner, T. R. Fischer, N. Farvardin, M. W. Marcellin, and R. H. Bamberger, "Comparison of different methods of classification in subband coding of images," *IEEE Trans. on Image Processing*, vol. 06, pp. 1473-1486, Nov. 1997.

55. N.S. Jayant and P. Noll, *Digital coding of waveforms: Principles and applications to speech and video*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

56. B.J. Kim and W.A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," *in Proceedings of Data Compression Conference*, pp. 251-260, 1997.

57. G.G. Langdon, "An Adaptive Run-length Coding Algorithm," *IBM Technical Disclosure Bulletin*, vol. 26, no. 7B, Dec. 1983.

58. Y. Linde, A. Buzo and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communications*, pp. 84-95, Jan. 1980.

59. W. Li, "Overview of fine granular scalability in MPEG-4 video standard," *IEEE Trans. on Circuits and Systems in Video Tech.*, vol. 11, no. 3, pp. 385-398, Mar. 2001.

60. A.S. Lewis and G. Knowles, "Image compression using 2-D wavelet transform," *IEEE Trans. on Image Processing*, vol. 1, no. 2, pp. 244-250, Apr. 1992.

61. J. Li and S. Lei, "Rate-distortion optimized embedding," *in Proceedings of Picture Coding Symposium*, pp. 201-206, Sept. 1997.

62. L. Luo, J. Li, S. Li, Z. Zhuang, and Y. Zhang, "Motion compensated lifting wavelet and its application in video coding," *in Proceedings of the IEEE International Conf. on Multimedia and Expo*, Aug. 2001.

63. S.P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. on Information Theory*, vol. 28, pp. 127-135, Mar. 1982.

64. J. Liu and P. Moulin, "Analysis of interscale and intrascale dependencies between image wavelet coefficients," *in Proceedings of IEEE International Conf. on Image Proc.*, pp. 669-671, Sept. 2000.

65. S. LoPresto, K. Ramchandran, and M. T. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," *in Proceedings of Data Compression Conference*, pp. 221-230, 1997.

66. S.G. Mallat, "A theory for multiresolution signal representation: The wavelet decomposition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, Jul. 1989.

67. J. Max, "Quantizing for minimum distortion," *IEEE Trans. on Information Theory*, vol. 6, pp. 7-12, Mar. 1960.

68. D.M. Monro, J.L. Aufranc, M.A. Bowers, and W.K. Poh, "Visual embedding of wavelet transform coefficients," *in Proceedings of the IEEE International Conf. on Image Proc.*, pp. 186 - 189, Sept. 2000.

69. D. Marpe and H. L. Cycon, "Very low bit-rate video coding using wavelet-based techniques," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 9, pp. 85-94, Feb. 1999.

70. D. Marpe and H.L. Cycon, "High-performance wavelet-based video coding using variable block-size motion compensation and adaptive arithmetic coding," *4th IASTED International Conference on Signal and Image Processing*, 2002.

71. M.W. Marcellin, M. J. Gormish, A. Bilgin, and M.P. Boliek, "An overview of JPEG-2000," *in Proceedings of the Data Compression Conference*, pp. 523-541, Mar. 2000.

72. S.A. Martucci, I. Sodagar, T. Chiang, and Y.Q. Zhang, "A zerotree wavelet video coder," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 7, no. 1, pp. 109-118, Feb. 1997.

73. Detlev Marpe, Thomas Wiegand, and Hans L. Cycon, "Design of a highly efficient wavelet-based video coding scheme," *in Processing of SPIE*, vol. 4671, pp. 1133-1141, Jan. 2002.

74. S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3397-3415, Dec. 1993.

75. MPEG-2 Test Model 5. *MPEG-2 encoder/decoder v.1.2*, available via http://www.mpeg.org/MPEG/MSSG/.

76. R. Neff and A. Zakhor, "Modulus quantization for matching pursuit video coding," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 10, no. 6, pp. 895-912, Sept. 2000.

77. R. Neff and A. Zakhor, "Matching Pursuit Video Coding - Part 1: Dictionary Approximation," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 12, no. 1, pp. 13-26, Jan. 2002.

78. R. Neff and A. Zakhor, "Very low bit rate video coding based on matching pursuits," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 7, pp. 158-171, Feb. 1997.

79. D.L. Neuhoff, "The other asymptotic theory of source coding," *in Joint IEEE/DIMACS Workshop on Coding and Quantization*, Oct. 1992.

80. Ortega, A. and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23-50, Nov. 1998.

81. A. Ortega, "Optimal rate allocation under multiple rate constraints," *in Proceedings of the Data Compression Conference*, Mar. 1996

82. A. Ortega and M. Vetterli, "Adaptive scalar quantization without side information," *IEEE Trans. on Image Processing*, vol. 6, pp. 665-676, May 1997.

83. E. Ordentlich, M. Weinberger, and G. Seroussi, "A low-complexity modeling approach for embedded coding of wavelet coefficients," *in Proceedings of Data Compression Conference*, pp. 408-417, 1998.

84. A.P. Petukhov, "Biorthogonal wavelet bases with rational masks and their application," Preprint, 1998.

85. W. Pennebaker and J. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, 1994.

86. D. Redmill, D. Bull, and P. Czerepinski, "Video coding using a fast non-separable matching pursuits algorithm," *in Proceedings of IEEE International Conf. on Image Proc.*, pp. 769-773, Oct. 1998.

87. R.C. Reininger and J.D. Gibson, "Distribution of the two-dimensional DCT coefficients for images," *IEEE Trans. on Communications*, vol. 31, no. 6, pp. 835-839, Jun. 1983.

88. B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. on Communication*, vol. 34, no. 11, pp. 1105-1115, Nov. 1986.

89. O. Rioul, "On the choice of wavelet filter for still image compression," *in Proceedings of IEEE ICASSP*, vol. 5, pp. 550-553, Apr. 1993.

90. O. Rioul, "Regular wavelets: a discrete-time approach," *IEEE Trans. on Signal Processing*, vol. 41, pp. 3572-3579, Dec. 1993.

91. E. A. Riskin, "Optimum bit allocation via the generalized BFOS algorithm," *IEEE Trans. on Information Theory*, vol. 37, pp. 400-402, Mar. 1991.

92. M.J. Riley and I.E.G. Richardson, *Digital Video Communication*, Artech House, 1997.

93. J.A. Robinson and Y. Shu, "Zerotree pattern coding of motion picture residues for error-resilient transmission of videos sequences", *IEEE Trans. Selected Areas in Comm.*, vol 18, no. 6, pp. 1099-1110, Jun 2000.

94. K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. on Image Processing*, vol. 2, no. 2, pp.160-176, Apr. 1993.

95. C.E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. Journal*, vol. 27, pp. 379-423, 1948.

96. Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1445-1453, Sept. 1988.

97. J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

98. V. Strela, P.N. Heller, G. Strang, P. Topiwala, and C. Heil, "The application of multiwavelet filter banks to image processing," *IEEE Trans. on Image Processing*, vol. 8, pp. 548-562, Apr. 1999.

99. T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 7, no. 1, Feb. 1997.

100. E. P. Simoncelli, "Modeling the joint statistics of images in the wavelet domain," *in SPIE Conference on Wavelet Applications in Signal and Image Processing*, pp. 206-214, Jul. 1999.

101. R. J. Safranek and J. D. Johnston, "A perceptually tuned subband image coder with image-dependent quantization and post-quantization data compression," *in Proceedings of IEEE ICASSP*, pp. 1945-1948, 1989.

102. O. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, and A. Zakhor, "Video compression using matching pursuits," *IEEE Trans. on Circuits and Systems for Video Tech.*, pp. 123-143, Feb. 1999.

103. G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.

104. A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Tech.*, pp. 243-250, Jun. 1996.

105. V. Strela, "Multiwavelets: Theory and applications," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 1996.

106. G.S. Sullivan, "Efficient scalar quantization of exponential and laplacian random variables," *IEEE Trans. on Information Theory*, vol. 42, pp. 1365-1374, Sept. 1996.

107. G.J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 74-90, Nov. 1998.

108. W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math Anal.*, vol. 29, no. 2, pp. 511-546, 1997.

109. D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. on Image Processing*, vol. 9, pp. 1158-1170, Jul. 2000.

110. C. Tu and T.D. Tran, "On context-based entropy coding of block transform coefficients," *in Proceedings of IEEE International Conf. on Image Proc.*, pp. 669-972, Sept. 2002.

111. M.J. Tsai, J. D. Villasenor, and F. Chen, "Stack-run image coding," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 6, no. 5, pp. 519-521, Oct. 1996.

112. D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. on Image Processing*, pp. 572-588, Sept. 1994.

113. P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ: Prentice Hall, 1992.

114. J.D. Villasenor, B. Belzer, and J. Liao, "Wavelet filter evaluation for image compression," *IEEE Trans. on Image Processing*, vol. 4, pp. 1053-1060, Aug. 1995.

115. J. Vass, B.B. Chai, and X. Zhuang, "Significance-linked wavelet video coder," *in Proceedings of IEEE ICASSP*, pp. 2829-2832, May 1998.

116. M. Vetterli and T. Kalker, "Matching pursuit for compression and application to motion compensated video coding," *in Proceedings of IEEE International Conf. on Image Proc.*, pp. 725-729, Nov. 1994.

117. M. Vetterli and J. Kovacevic, *Wavelet and Subband Coding*, Englewood Cliffs, NJ: Prentice Hall, 1995.

118. L. Vandendorpe and B. Macq, "Optimum quantization for subband coders," *in Proceedings of Visual Communications and Image Processing*, vol. 1360, pp. 898-908, 1990.

119. C. Vleeschouwer and B. Macq, "New dictionaries for matching pursuit video coding," *in Proceeding of the IEEE International Conf. on Image Proc.*, pp. 764-768, 1998.

120. G.K. Wallace, "The JPEG Still Picture Compression Standard," *Comm. of the ACM*, vol. 34, no. 4, pp. 31-44, Apr. 1991.

121. A.B. Watson, "Perceptual optimization of DCT color quantization matrices," *in Proceeding of the IEEE International Conf. on Image Proc.*, pp. 100-104, Nov. 1994.

122. D. Wang, C. Labit, and J. Ronsin, "Segmentation-based motion-compensated video coding using morphological filters," *IEEE Trans. on Circuits and Systems for Video Tech.*, no. 3, pp. 549-554, Jun. 1997.

123. F. Wu, S. Li, and Y.Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 11, no. 3, pp. 332-344, Mar. 2001.

124. I.H. Witten, R.M. Neal, and J.G. Cleary, "Arithmetic coding for data compression," *Comm. of the ACM*, vol. 30, pp. 520-540, Jun. 1987.

125. R.C. Wood, "On optimum quantization," *IEEE Trans. on Information Theory*, vol. 15, pp. 248-252, Mar. 1969.

126. M. Weinberger, G. Seroussi, G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS", *IEEE Trans. Image Processing*, vol. 9, pp.1309-1324, Aug. 2000.

127. X. Wu, "High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression," *in Proceedings of Thirty-First Asilomar Conf. on Signals, Systems and Computers*, vol. 2, pp. 1378-1382, 1997.

128. J. Wen and J. Villasenor, "A class of reversible variable length codes for robust image and video coding," *in Proceedings of the IEEE International Conf. on Image Proc.*, vol. 2, pp. 65-68, Oct. 1997.

129. A. B. Watson, G. Y. Yang, J. A. Solomon, and J. Villasenor, "Visibility of wavelet quantization noise," *IEEE Trans. on Image Processing*, vol. 6, pp. 1164-1175, Aug. 1997.

130. Z. Xiong, K. Ramchandran, and M.T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. on Image Processing*, vol. 6, pp. 677-693, May 1997.

131. Z. Xiong, K. Ramchandran, and M.T. Orchard, "Wavelet packet image coding using space-frequency quantization," *IEEE Trans. on Image Processing*, vol. 7, pp. 892-898, Jun 1998.

132. Z. Xiong, K. Ramchandran, M.T. Orchard, and Y.Q. Zhang "A comparative study of DCT- and Wavelet-based image coding," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 9, no. 5, Aug. 1999.

133. Z. Xiong and X. Wu, "Wavelet image coding using trellis coded space-frequency quantization," *IEEE Signal Processing Letters*, vol. 6, no. 7, Jul 1999.

134. E.Y. Lam and J.W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. on Image Processing*, vol. 9, no. 10, Oct. 2000.

135. Y.Yoo, A.Ortega, and B.Yu, "Adaptive quantization of image subbands with efficient overhead rate distortion," *in Proceedings of the IEEE International Conf. on Image Proc.*, vol. 2, pp. 361-364, Sept. 1996.

136. X. Yang and K. Ramchandran, "Hierarchical backward motion compensation for wavelet video coding using optimized interpolation filters," *in Proceedings of the IEEE International Conf. on Image Proc.*, pp. 85-88, Oct. 1997.

137. Y.Q. Zhang and S. Zafar, "Motion compensated wavelet transform coding for color video compression," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 2, pp.285-296, 1992.